TECHNISCHE UNIVERSITÄT WIEN

ACIN AUTOMATION & CONTROL INSTITUTE INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

Dynamisches Greifen von 3D-Objekten mittels robotischem System

DIPLOMARBEIT

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom-Ingenieurs (Dipl.-Ing.)

unter der Leitung von

Univ.-Prof. Dr. techn. A. Kugi
Univ.-Ass. Dipl.-Ing. C. Unger

eingereicht an der

Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik
Institut für Automatisierungs- und Regelungstechnik

von
Grander Florian
Matrikelnummer 1426462
[redacted]

Wien, Juni 2021

Vorwort

Neben meiner Arbeit am Institut für Automatisierungs- und Regelungstechnik habe ich etliche Roboteranwendungen beobachten dürfen, wodurch mein Interesse und Wunsch für ein eigenes Projekt an einem Industrieroboter geweckt wurde. Weiters hege ich ein großes Interesse für Künstliche Intelligenz und Machine Vision Anwendungen.

Hiermit möchte ich mich recht herzlich bei allen, die bei der Entstehung dieser Masterarbeit beteiligt waren, bedanken. Ein besondere Dank geht an meinen Betreuer Dipl.-Ing. Christoph Unger, sowie an Dipl.-Ing. Dr.techn. Christian Hartl-Nesic, Dipl.-Ing. Florian Beck, und MSc. Minh Nhat VU für ihre Unterstützung, die motivierenden Gespräche und das zeitaufwendige Korrekturlesen. Ein spezieller Dank geht an Univ.-Prof. Dr.techn. Andreas Kugi für die Erlaubnis dieses Projekt umsetzen zu dürfen und an das ganze Institut für die nette Atmosphäre.

Zum Schluss bedanke ich mich bei all meinen Freunden, meiner Familie und ganz speziell auch bei meiner Partnerin Kristina für Ihre Unterstützung und motivierenden Worte während meiner gesamten Studienzeit.

Wien, Juni 2021

Abstract

In this diploma thesis, a Schunk SDH2 gripper hand is used as an end effector on a KUKA LBR iiwa 14 R820 industrial robot. A Basler avA1000-gc industrial camera is mounted next to the gripper. The pose of a known object from the individual images taken by the camera is determined by optical object recognition. The goal is to grasp the optically detected object with the gripper. For this purpose, the dynamic model of the robot extended by the gripper and the camera is derived. Additionally, in order to plan a grasp and thus the finger motion, further state variables of the grasp are defined. For an unambiguous conversion between the gripper and the grasp states, the forward and backward kinematics are derived.

For a successful grasping of the object, the motion of the robot and the fingers are planned first in an offline trajectory planning. In order to react to any errors and also updates of the pose estimation of the grasping object during the robot motion, a fast online trajectory replanning is implemented.

The planned trajectory of the robot serves for the feedforward controller and a trajectory tracking controller accounts for the trajectory errors. Feedforward and feedback control are also implemented for the grasping hand. Several experiments are performed and the results are presented and discussed.

Kurzzusammenfassung

In dieser Arbeit wird auf einem KUKA LBR iiwa 14 R820 Industrieroboter eine Schunk SDH2 Greifhand als Endeffektor montiert. Neben der Greifhand ist eine Basler avA1000-gc Industriekamera angebracht. Mittels optischer Objekterkennung wird aus den einzelnen Bildern der Kamera von einem bekannten Objekt die Pose bestimmt. Ziel ist es, das optisch detektierte Objekt mit der Greifhand zu greifen. Dazu wird zunächst für den um die Greifhand und die Kamera erweiterten Roboter das dynamische Modell hergeleitet. Um einen Griff und damit die Fingerbewegung zu planen, werden weitere Zustandsgrößen des Griffs definiert. Für eine eindeutige Umrechnung zwischen den Zuständen der Greifhand und des Griffs werden die Vorwärts- und die Rückwärtskinematik hergeleitet.

Für ein erfolgreiches Greifen des Objekts wird zunächst in einer Offline-Trajektorienplanung die Bewegung des Roboters und der Finger geplant. Um während der Roboterbewegung auf etwaige Fehler bzw. Updates der Posenschätzung des Greifobjekts zu reagieren, wird eine schnelle Online-Trajektorienplanung implementiert.

Die geplante Trajektorie des Roboters wird während des Versuchs vorgesteuert und ein Trajektorienfolgeregler regelt die Trajektorienfehler. Für die Greifhand wird ebenfalls eine Vorsteuerung und ein Regler implementiert. Es werden mehrere experimentelle Versuche durchgeführt und die Ergebnisse dargestellt und diskutiert.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Motivation und Einleitung

Durch den steigenden Grad der Automatisierung werden in der Industrie immer mehr Roboter eingesetzt. Gleichbleibend hohe Produktionsqualität, sicherere Arbeitsabläufe und somit geringeres Verletzungsrisiko, effizientere Produktion und langfristige Senkung der Produktionskosten sind die Treiber für den Einsatz von Industrierobotern. Eine große Herausforderung für Unternehmen der heutigen Zeit ist die wachsende Anzahl an unterschiedlichen Produkten in immer kürzer werdenden Zeitintervallen. Aufgrund dessen geht ein Trend in Richtung eines modularen Produktionsaufbaus, in dem sich nicht benötigte Komponenten einfach zu- und abschalten lassen. In diesem flexiblen Setting spielen Roboter eine wesentliche Rolle, da sie einfach austauschbar sind und schnell auf neue Produkte bzw. Produktionsprozesse angepasst werden können. Für robotergestützte Anwendungen wird grundsätzlich vorausgesetzt, dass das Werkstück immer an der selben Position und der selben Orientierung dem Arbeitsschritt zugeführt wird. Dadurch wird der Einsatz dieser Systeme erheblich erleichtert, da sie durch einen einfachen Teach-In-Prozess programmiert werden können. Das Zuführen des Werkstücks im jeweiligen Arbeitsschritt in zufälliger Position und Orientierung steigert die Komplexität und den Programmieraufwand erheblich, erhöht jedoch die Flexibilität der Produktionslinie. Zusätzlich kann durch Optimierung des Greifvorgangs bzw. durch dynamisches Greifen die Zykluszeit des Produktionsprozesses verringert und somit die Produktionsrate gesteigert werden. Die Detektion und die Bestimmung der Pose und die Planung der Roboterbewegung kann unter Umständen sehr aufwendig sein.

Machine Vision hat sich in den vergangenen Jahren durch den Einsatz Künstlicher Intelligenz rasant weiterentwickelt. Auf Basis von Convolutional Neural Networks, kurz CNNs, ist es mittlerweile möglich Objekte in Bildern zu detektieren und deren Pose in Relation zur Kamera zu bestimmen.

In [1] werden aus einem einzelnen Kamerabild für alle erkannten Objekte die acht Punkte der Bounding-Box sowie der Mittelpunkt geschätzt. Aus diesen neun Punkten wird anschließend über den EPnP (Efficient Perspective-n-Point) Algorithmus [2] die Pose berechnet.

In [3] werden mit einem Deep-CNN aus einem RGB-Bild zunächst Featuremaps generiert. Ein weiteres Netzwerk bestimmt die Bereiche im Bild, in denen Objekte abgebildet sind. Mit den Features aus den detektierten Bereichen werden über ein weiteres neuronales Netzwerk direkt die Posen der Objekte bestimmt.

In [4] werden für alle Pixel des Bildes Vektoren generiert, die zu sog. Keypoints am Objekt zeigen. Anhand dieser Keypoints wird weiters mittels EPnP-Algorithmus die Pose bestimmt.

Die genannten Methoden bestehen entweder aus sehr großen Netzwerken, oder es werden mehrere Stufen durchlaufen, was die Inferenzzeit erhöht und somit eine Echtzeitimplementierung kaum möglich macht. Eine sehr effiziente Methode wird in [5] vorgestellt.

Dort werden aufbauend auf Standardnetzwerken für Real-Time Object-Detection YOLO (you only look once) erneut Keypoints für jedes Objekt detektiert und mittels EPnP-Algorithmus die Pose bestimmt.

Das Trainieren solcher Netzwerke benötigt eine große Anzahl annotierter Trainingsdaten in Form von Bildern. Üblicherweise werden die Trainingsdaten mit großem Aufwand händisch annotiert. In [6] wird eine Methode vorgestellt, mit welcher diese Trainingsdaten künstlich und in relativ kurzer Zeit erstellt werden können. Dabei handelt es sich um eine Erweiterung für die Unreal Grafikengine, mit welcher zufällige Bilder mit den gewünschten Objekten generiert und annotiert werden. Damit können Trainingsdaten für alle typischen neuronalen Netze erstellt werden. Nachteil dieser Methode ist, dass es sich bei den Trainingsdaten um künstliche Bilder handelt und etwas von den Lichtverhältnissen und Rauscheigenschaften von realen Bildern abweicht.

Mit diesen Methoden ergeben sich ganz neue Möglichkeiten im Bereich der Robotik. Die Anwendungen für die online Bestimmung der Pose mittels RGB-Kameras lassen sich in die Teilbereiche *eye-in-hand* und *eye-to-hand* unterteilen. Bei den *eye-in-hand* Systemen, solche Systeme werden auch als *visual servoing* Systeme bezeichnet, befindet sich die Kamera üblicherweise am Endeffektor des Roboters, ist somit in Bewegung und liefert eine detailliertere Darstellung der Szene. Bei *eye-to-hand* Systemen ist die Kamera stationär montiert und auf den Roboter gerichtet, womit eine wenig präzisere Darstellung, aber eine bessere globale Übersicht erreicht wird. Die Kombination beider Systeme samt ihren Vorteilen findet auch ihre Anwendungen, was die Komplexität jedoch wesentlich erhöht. Die Arbeit [7] beschäftigt sich z. B. mit einem Setup nach dem *eye-in-hand* Prinzip, bei dem Werkteile von Paletten gesammelt werden. Diese Diplomarbeit richtet sich ebenfalls nach dem *eye-in-hand* Prinzip.

Roboter werden hauptsächlich stationär in die Industrieanlage integriert. Je nach Anwendungsfall bzw. Fertigungsprozess, z. B. dem Entnehmen eines fertigen Bauteils aus einer CNC-Maschine und einsetzen eines neuen Rohlings, ergeben sich für den Roboter erhebliche Wartezeiten. Ein Roboter auf einer mobilen Plattform, der zwischen den einzelnen Fertigungsmaschinen wechselt, ist wesentlich effizienter und kostengünstiger im Vergleich zu einem Roboter an jeder Fertigungsmaschine. In [8] wird ein Roboter auf einer mobilen Plattform vorgestellt. Die Grobpositionierung des Roboters erfolgt über die Plattform, mittels Kameras wird eine Feinpositionierung ausgeführt. In [9] wird ein weiterer Roboter auf einer mobilen Plattform vorgestellt.

Eine Herausforderung ist weiterhin die Trajektorienplanung. Dabei wird zwischen Offline- und Online-Trajektorienplanung unterschieden. Die Offline-Trajektorienplanung findet vor der eigentlichen Roboterbewegung, also *offline*, statt. Im Gegensatz dazu passiert die Online-Trajektorienplanung während der Ausführung, also *online*. Im Vergleich zur Offline-Trajektorienplanung sind bei der Online-Trajektorienplanung die Anforderungen an die Hardware und der Rechenaufwand wesentlich größer. In [10, 11] werden Methoden für eine schnelle Online-Neuberechnung der Trajektorie vorgestellt. In [10] wird für einen Portalkran der kontinuierliche Start- und der Zielbereich jeweils in ein Gitter mit äquidistantem Abstand zwischen den Gitterpunkten diskretisiert. Von jedem Gitterpunkt im Startbereich wird zunächst in einer Offline-Trajektorienplanung eine optimale Trajektorie zu jedem Gitterpunkt im Zielbereich berechnet und in einer Datenbank gespeichert. Für Start- und Zielkonfigurationen zwischen den Gitterpunkten wird die optimale Trajektorie mit

dem geringsten Abstand zu den Gitterpunkten gewählt und über eine schnelle Online-Trajektorienplanung ein Update der Trajektorie aus der Datenbank ausgeführt. In [11] wird diese Methode adaptiert, um ein inverses Pendel mit einem KUKA Industrieroboter aufzuschwingen. In beiden Arbeiten existiert ein flacher Ausgang, somit vereinfacht sich für die Trajektorienplanung das nichtlineare dynamische System jeweils zu einem linearen Integratorsystem.

In dieser Arbeit wird aufbauend auf [10, 11] für ein bekanntes Objekt, dessen Pose optisch mittels einer Kamera gemessen bzw. geschätzt wird, offline eine Trajektorie berechnet. Während der Roboterbewegung wird die Pose zyklisch neu gemessen und durch eine schnelle Online-Trajektorienplanung eine neue Trajektorie berechnet. Für das Greifen wird die Greifhand Schunk SDH2 als 2-Finger-Greifer verwendet. Für die Finger wird ein kinematisches Modell hergeleitet und eine Trajektorie der Finger für ein erfolgreiches Greifen berechnet.

1.1 Gliederung der Arbeit

Ziel ist das dynamische Greifen eines Objekts mittels des Industrieroboters KUKA LBR iiwa 14 R820. Die dazu notwendigen Schritte lassen sich in einzelne Kategorien aufteilen.

Im ersten Teil von Kapitel 2 werden die Komponenten vorgestellt, welche für die Durchführung des dynamischen Greifvorgangs verwendet werden. Der zweite und dritte Teil beschäftigt sich mit der mathematischen Modellierung des Roboters und der Greifhand.

Kapitel 3 beinhaltet die optische Objekterkennung bzw. Posenschätzung aus den einzelnen Bildern der Kamera. Um den Einfluss von Ausreißern (Outlier) zu verringern und gleichzeitig eine bessere Posenschätzung zwischen den einzelnen Bildern zu ermöglichen, wird ein Kalman-Filter auf Basis eines Kalman-Filters zur Pose-Estimation mittels IMU entworfen.

Kapitel 4 beinhaltet die Offline- und Online-Trajektorienplanung des Roboters und der Finger für ein erfolgreiches Greifen des Objekts.

In Kapitel 5 werden der Trajektorienfolgeregler des Roboters und der Greifhand entworfen.

Die Implementierung des kompletten Setups und die Ergebnisse einiger experimenteller Versuche sind in Kapitel 6 dargestellt.

2 Modellbildung

Dieses Kapitel beschäftigt sich mit der Systembeschreibung und der Herleitung der mathematischen Modelle. Mit den berechneten Modellen können Simulationen durchgeführt werden bzw. dienen sie als Basis für die Trajektorienplanung in Kapitel 4 sowie für den Entwurf eines Trajektorienfolgereglers in Kapitel 5.

Im ersten Teil dieses Kapitels werden die verwendeten Komponenten näher vorgestellt und gezeigt wie diese zu einem Setup zusammengebaut werden.

In Abschnitt 2.2 und 2.3 werden die mathematischen Modelle für den Roboter und für die Greifhand hergeleitet. Für das dynamische Modell des Roboters werden die Greifhand und die Kamera als Punktmasse betrachtet und der Roboter um diese erweitert.

Um das Greifen eines Objekts mit den Fingern zu planen, wird zunächst zwischen den Fingerspitzen ein Greifpunkt definiert. Damit ist jedoch nur die Position des Greifpunktes gegeben, womit noch keine Planung eines Griffs möglich ist. Mit der Einführung weiterer Zustandsgrößen für die Öffnung der Finger und die Orientierung der Tangente der Greifflächen bei geschlossenem Griff vervollständigt sich der Greifzustand und ein Griff kann geplant werden. Über die Vorwärtskinematik wird der Greifzustand aus den Zuständen der Finger berechnet. Umgekehrt werden über die Rückwärtskinematik die Zustände der Finger aus den Greifzuständen berechnet. Die Herleitung der analytischen Lösung der Vorwärts- bzw. Rückwärtskinematik befindet sich im zweiten Teil dieses Kapitels. Für die Rückwärtskinematik muss ein zulässiges Gebiet definiert werden, da die Lösung der Rückwärtskinematik nicht für alle möglichen Greifzustände existiert. Durch die interne Steuerung bzw. Regelung der Greifhand, auf die später in dieser Arbeit in Kapitel 5 und 6 näher eingegangen wird, ist die Kinematik ausreichend und auf ein dynamisches Modell wird verzichtet.

2.1 Systembeschreibung

In diesem Abschnitt werden die verwendeten Komponenten der Roboter, die Greifhand, die Kamera und das verwendete Greifobjekt näher vorgestellt. In Abschnitt 2.1.1 wird der verwendete Roboter KUKA LBR iiwa 14 R820 und die für die Modellierung verwendeten Koordinatensysteme $(0_i x_i y_i z_i), i \in \{1, \ldots, 7\}$ vorgestellt. Abschnitt 2.1.2 beschäftigt sich mit der geometrischen Beschreibung des verwendeten Greifers Schunk SDH2. Abschnitt 2.1.3 zeigt das zu greifende Objekt. Das in dieser Arbeit verwendete Kamerasystem wird in Abschnitt 2.1.4 vorgestellt. Für den Versuch werden die Greifhand und die Kamera, wie in Abschnitt 2.1.5 gezeigt, als Endeffektor am Roboter montiert.

2.1.1 Roboter Kuka LBR iiwa 14 R820

Der verwendete Roboter KUKA LBR iiwa 14 R820 ist in Abbildung 2.1 dargestellt. Der Roboter besteht aus 8 Starrkörperelementen, welche über elastische Drehgelenke miteinander verbunden sind. Die Koordinatensysteme $(0_i x_i y_i z_i), i \in \{1, \ldots, 7\}$ sind in den Drehgelenken platziert. Die Drehung erfolgt jeweils um die z-Achse. Somit verfügt der Roboter über sieben Freiheitsgrade, welche in weiterer Folge mit $q_i, i \in \{1, \ldots, 7\}$ bezeichnet werden. Das Koordinatensystem $(0_0 x_0 y_0 z_0)$ ist die Roboterbasis und stellt gleichzeitig das Basiskoordinatensystem $(0xyz)$ dar. Das Koordinatensystem $(0_e x_e y_e z_e)$ ist die Stelle, an welcher der Endeffektor montiert wird. Zusätzlich zur Drehwinkelmessung verfügt jedes Gelenk über einen Drehmomentsensor, mit welchem das Drehmoment zwischen Motorachse und Gelenksachse gemessen wird. Damit können von außen einwirkende Kräfte erkannt und darauf reagiert werden. Des Weiteren lässt sich auf Basis von [12] die Singuläre Störtheorie anwenden, um ein Regelungskonzept zu entwickeln. Dabei beschreibt das langsame Teilsystem die Dynamik des Starrkörpersystems des Roboters und das schnelle Teilsystem die Dynamik der flexiblen Gelenke. Die benötigten Parameter des Roboters sind in Tabelle A.2 im Anhang angegeben.

2.1.2 Greifhand Schunk SDH2

Als Endeffektor des Roboters wird die in Abbildung 2.2 abgebildete Greifhand Schunk SDH2 mit drei Fingern verwendet. Jeder Finger besteht aus einem proximalen und einem distalen Element, welche durch ein Drehgelenk miteinander verbunden sind. Das proximale Element ist über ein weiteres Drehgelenk mit der Basis verbunden, wodurch sich zwei Freiheitsgrade pro Finger ergeben. Durch einen weiteren Freiheitsgrad ist eine Rotation von Finger 1 und 3 um die Hochachse möglich. Finger 1 und 3 drehen sich dabei synchron, aber in entgegengesetzter Richtung. In Abbildung 2.2 sind die körperfesten Koordinatensysteme der Basis $(0_h x_h y_h z_h)$ und der Finger $(0_{hi} x_{hi} y_{hi} z_{hi})$, $i \in \{1, \ldots, 6\}$ inklusive der Freiheitsgrade $q_{h1} \in [0°, 90°]$ und $q_{hi} \in [-90°, 90°]$, $i \in \{2, \ldots, 7\}$ eingezeichnet. Die gleichzeitige Drehung der Finger 1 und 3 um die Hochachse erfolgt jeweils um die z-Achse des jeweiligen Koordinatensystems. Die Drehung für die Drehgelenke erfolgt jeweils um die x-Achse des jeweiligen Koordinatensystems.

Zum Greifen eines Versuchsobjekts werden in dieser Arbeit nur Finger 1 und Finger 3 benötigt. Finger 2 bleibt auf der initialen Konfiguration $q_{h4,0} = 90°$ und $q_{h5,0} = 60°$ und ist in den restlichen Abbildungen dieser Arbeit nicht mehr dargestellt. Die erforderlichen Parameter sind in Abbildung 2.3 eingezeichnet und in Tabelle A.3 angegeben.

2.1.3 Greifobjekt

Als Versuchsobjekt wird das in Abbildung 2.4 dargestellte Blechbiegeteil verwendet. Die Basis des Greifobjekts ist das Koordinatensystem $(0_o x_o y_o z_o)$. Weiters wird ein Greifpunkt definiert, dargestellt durch das Koordinatensystem $(0_g x_g y_g z_g)$. Für die Wahl des Greifpunktes existieren etliche Algorithmen, die aus den geometrischen Eigenschaften des Objekts und den unterschiedlichsten Bewertungskriterien hinsichtlich der Qualität eines Griffs den optimalen Greifpunkt berechnen, siehe dazu [13, 14]. Eine Methode ermöglicht z. B. die Berechnung der optimalen Positionen der Kontaktpunkte zwischen Finger und

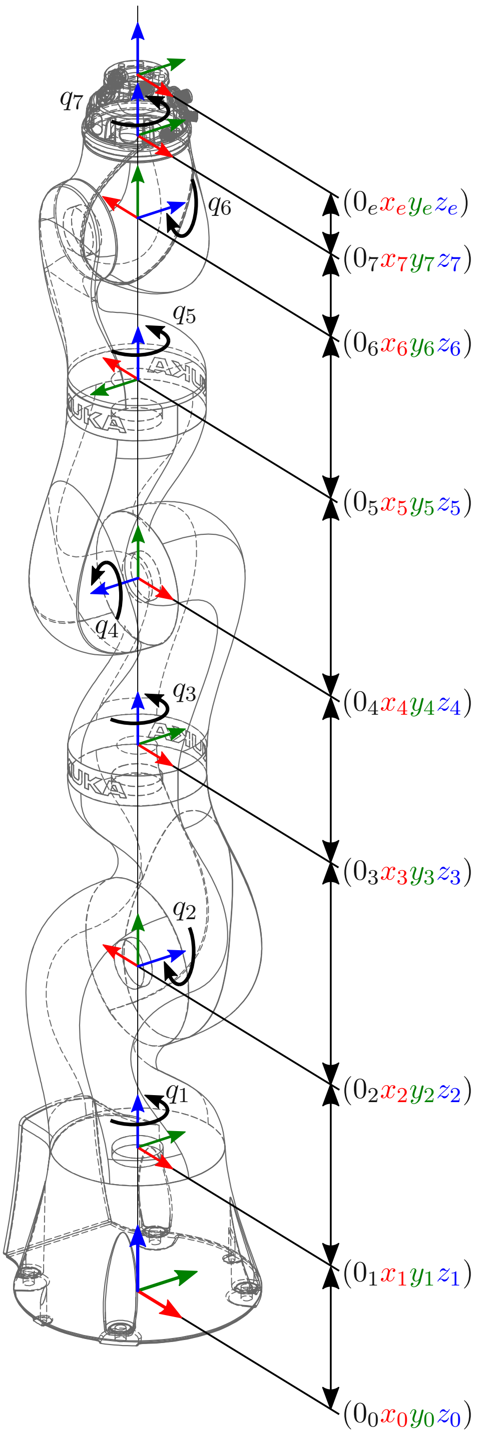


Abbildung 2.1: Roboter KUKA LBR iiwa 14 R820 mit eingezeichneten Koordinatensystemen $(0_i x_i y_i z_i)$, $i \in \{0,\ldots,7,e\}$ und den Freiheitsgraden q_i, $i \in \{1,\ldots,7\}$.

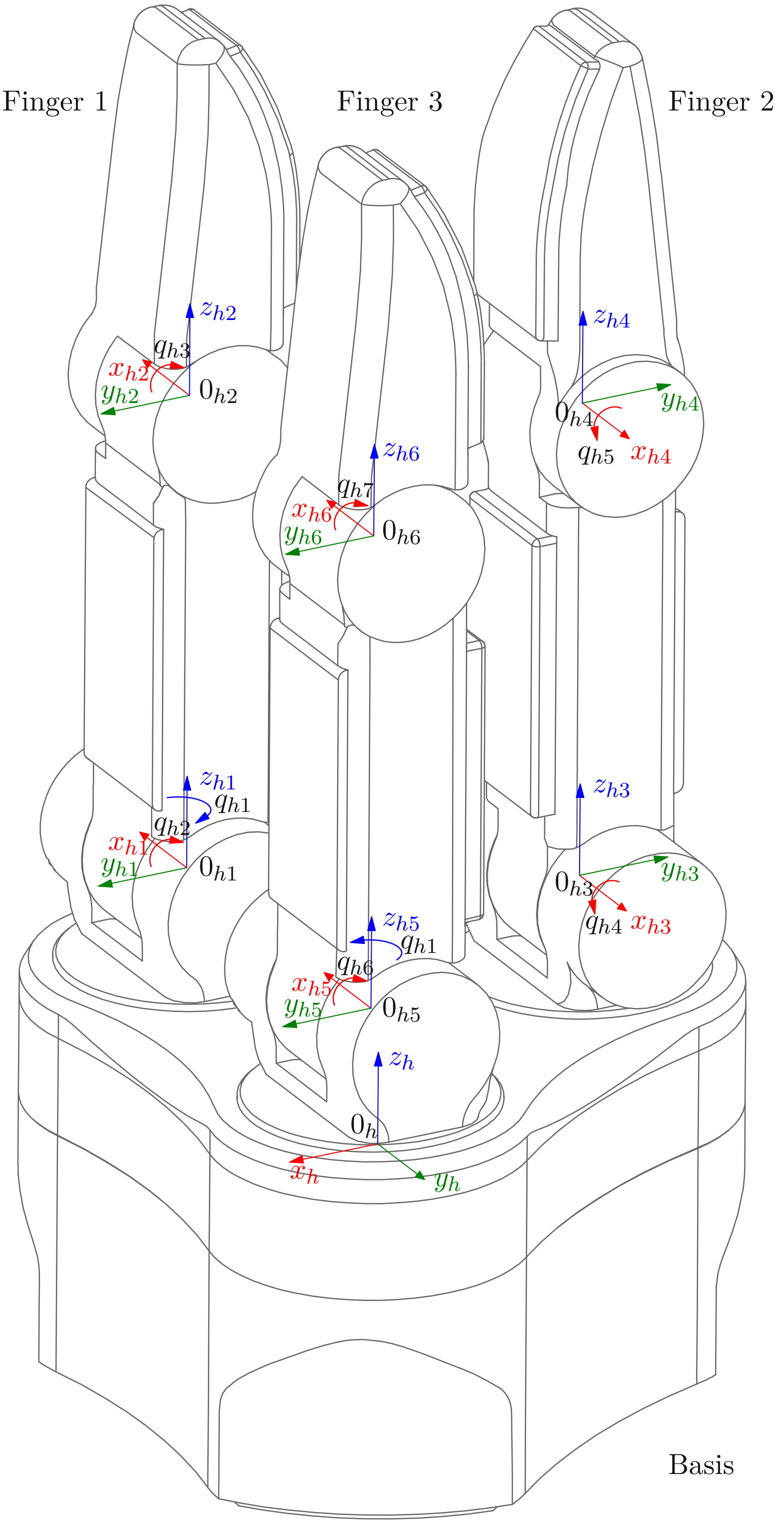


Abbildung 2.2: Greifhand Schunk SDH2 mit eingezeichneten Koordinatensystemen $(0_h x_h y_h z_h)$, $(0_{hi} x_{hi} y_{hi} z_{hi})$, $i \in \{1, \ldots, 6\}$ und den Freiheitsgraden q_{hi}, $i \in \{1, \ldots, 7\}$.

 2 Modellbildung

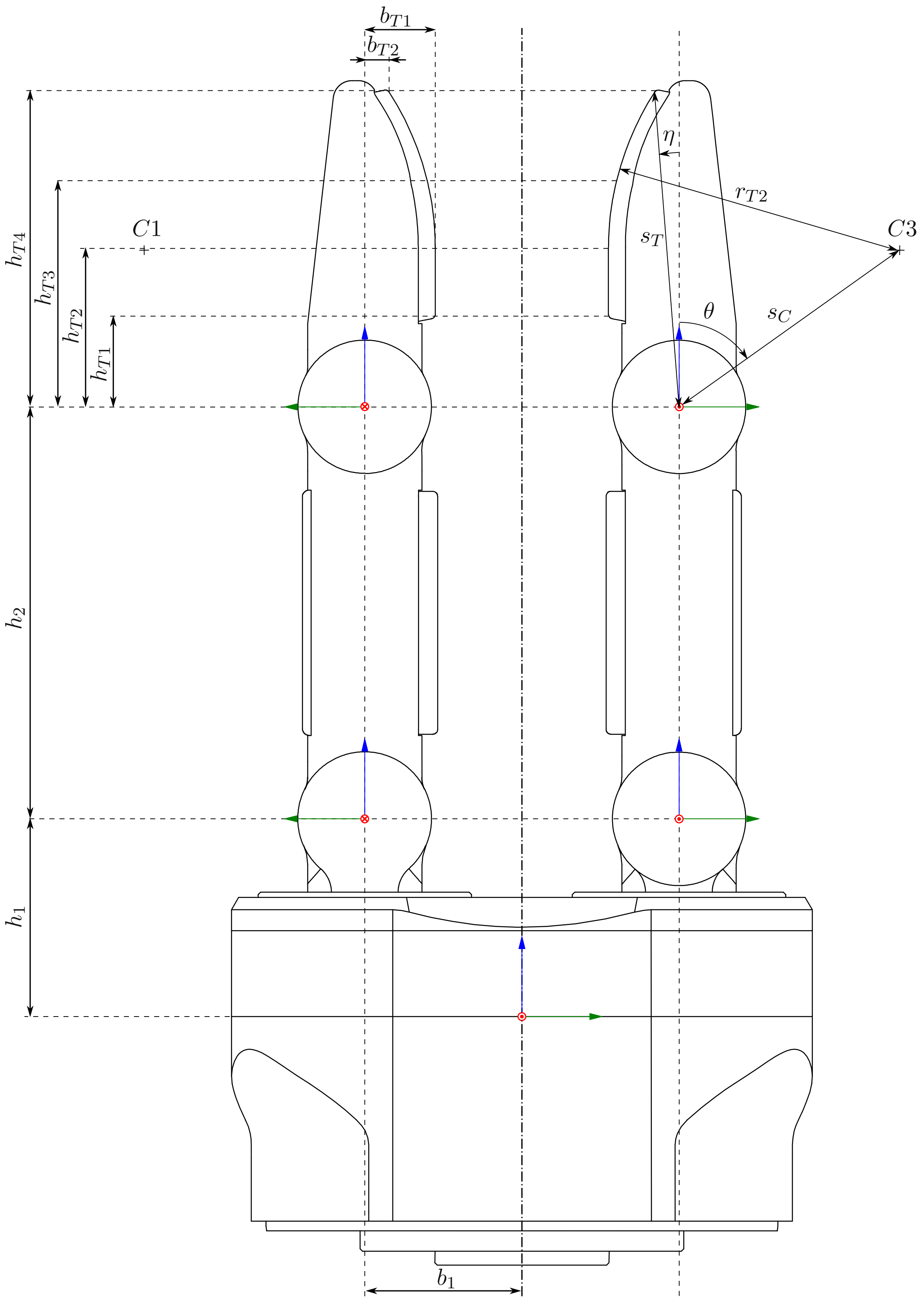


Abbildung 2.3: Seitenansicht Schunk SDH2 ohne Finger 2 für $q_{h1} = 90°$ mit eingezeichneten Parametern.

Objekt, so dass die wirkenden Klemmkräfte maximiert werden. Eine weitere Methode ermöglicht eine Berechnung der Kontaktpunkte, so dass die Widerstandsfähigkeit durch externe auf das Objekt einwirkende Kräfte aufgrund der Zentrifugal- bzw. Corioliskraft oder einer sonstigen externen Störung maximiert wird. Doch die reine Bestimmung der optimalen Kontaktpunkte ist nicht ausreichend. Die Geometrie der Finger des Greifers muss ebenfalls berücksichtigt werden, um sicherzustellen, dass keine Kollisionen auftreten. Der Einfachheit halber wurde auf eine Implementierung dieser Algorithmen in dieser Arbeit verzichtet und manuell ein statischer Greifpunkt definiert.

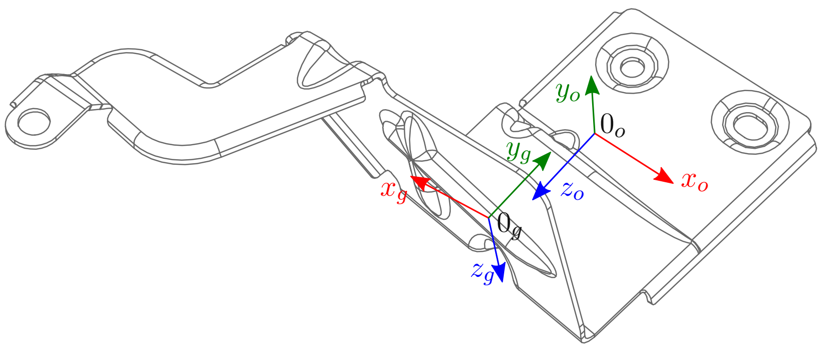


Abbildung 2.4: Greifobjekt mit eingezeichneten Koordinatensystemen ($0_o x_o y_o z_o$) für die Objektbasis und ($0_g x_g y_g z_g$) für den Greifpunkt.

2.1.4 Kamera Basler avA1000

Für die Objekterkennung wird die GigE-Kamera Basler avA1000-100gc von Abbildung 2.5 verwendet. Sie liefert Bilder mit einer Auflösung von 1024x1024 Pixel mit einer maximalen Bildrate von 100 fps. Die Kamera hat einen globalen Shutter. Dabei werden alle Zeilen eines Bildes gleichzeitig belichtet, dadurch entstehen keine Verzerrungen bei sich bewegenden Objekten. Durch die hohe maximale Bildrate sind kurze Belichtungszeiten möglich. Dadurch sind schnellere Bewegungen der Kamera möglich, bevor das Bild durch die Bewegungsunschärfe verschwommen dargestellt und die Objekterkennung erschwert wird. Neben den Standard RGB-Kameras existieren auch RGBD-Kameras, bei denen zusätzlich zum RGB-Bild über einen weiteren Kanal auch die Tiefeninformationen zur Verfügung stehen. Bei diesen Kamerasystemen ist die Bildrate der RGB-Bilder üblicherweise auf 30 fps limitiert.

2.1.5 Versuchsaufbau

Die Greifhand und die Kamera werden nach dem Eye-in-Hand Prinzip, wie in Abbildung 2.6 dargestellt, als Endeffektor am Roboter montiert. Durch die Initialkonfiguration des Roboters ist die Kamera auf einen definierten Bereich ausgerichtet, in dem das zu greifende Objekt aus Abbildung 2.4 platziert wird. Mittels Objekterkennung wird die Translation

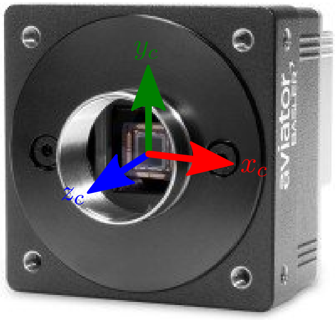


Abbildung 2.5: Verwendete Kamera Basler avA1000-100gc mit dem Basis Koordinatensystem $(0_c x_c y_c z_c)$.

und Rotation zwischen dem Koordinatensystem der Kamera $(0_c x_c y_c z_c)$ und jenem des Greifobjekts $(0_o x_o y_o z_o)$ geschätzt. Ausgehend von der geschätzten Pose, dem dynamischen Modell des Roboters, dem eingeführten Greifzustand, sowie dem kinematischen Modell für den Greifpunkt erfolgt die Planung für den dynamischen Greifvorgang. Der zweite Finger der Greifhand wird durch das Design des Greifobjekts und die Position des Greifpunktes nicht benötigt. Die Greifhand wird somit als ein 2-Finger Greifer eingesetzt.

2.2 Roboter mit Greifhand und Kamera

Im folgenden Abschnitt wird das mathematische Modell des Aufbaus aus Abbildung 2.6 hergeleitet. Die Kamera und die Greifhand werden dabei als Punktmasse modelliert. Die genauen Werte der Trägheitsmomente sind nicht bekannt und wurden jeweils über einen einzelnen Volumenkörper mit konstanter Massenverteilung abgeschätzt.

2.2.1 Kinematisches Modell

Der Roboter stellt eine Kombination von Starrkörpern dar, welche über Drehgelenke miteinander verbunden sind. In jedem Gelenk wird, wie in Abbildung 2.1 dargestellt, ein Koordinatensystem platziert. Das kinematische Modell beschreibt die Pose des Koordinatensystems der Greifhand $(0_h x_h y_h z_h)$ und der Kamera $(0_c x_c y_c z_c)$ bezogen auf das Basiskoordinatensystem $(0xyz)$ in Abhängigkeit der Gelenkswinkel des Roboters $\mathbf{q}^{\mathrm{T}} = [q_1, \ldots, q_7]$.

Eine Rotation $\mathbf{R}_i^j$ zwischen zwei Koordinatensystemen $(0_i x_i y_i z_i)$ und $(0_j x_j y_j z_j)$ ist über

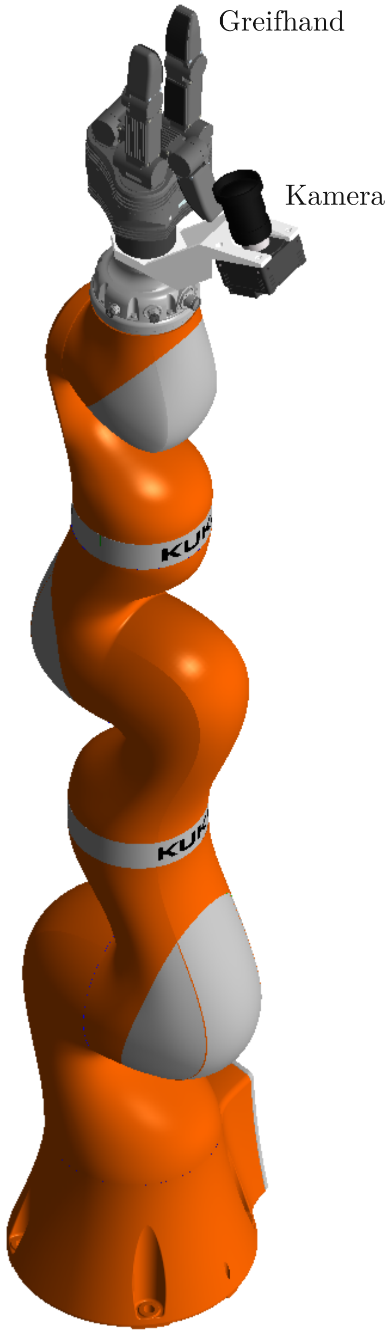


Abbildung 2.6: Versuchsaufbau des Roboters mit montierter Kamera und Greifhand als Endeffektor.

Multiplikationen der Rotationsmatrizen

$$\mathbf{R}_{z,\phi} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.1a}$$

$$\mathbf{R}_{y,\theta} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \tag{2.1b}$$

$$\mathbf{R}_{x,\psi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix} \tag{2.1c}$$

definiert [15]. Dabei bezeichnet $\mathbf{R}_{z,\phi}$ eine Rotation um den Winkel ϕ um die z-Achse, $\mathbf{R}_{y,\theta}$ eine Rotation um den Winkel θ um die y-Achse und $\mathbf{R}_{x,\psi}$ eine Rotation um den Winkel ψ um die x-Achse. Die Translation zwischen zwei Koordinatensystemen $(0_i x_i y_i z_i)$ und $(0_j x_j y_j z_j)$ ist über den Verschiebungsvektor $\mathbf{d}_i^j$ gegeben. Die Vorwärtskinematik wird über die homogene Transformation hergeleitet. Dabei lassen sich Rotationen und Translationen zwischen Koordinatensystemen $(0_i x_i y_i z_i)$ und $(0_j x_j y_j z_j)$ durch die homogene Transformation

$$\mathbf{H}_i^j = \begin{bmatrix} \mathbf{R}_i^j & \mathbf{d}_i^j \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \tag{2.2}$$

angeben. Die inverse homogene Transformation ist über

$$\mathbf{H}_j^i = \left(\mathbf{H}_i^j\right)^{-1} = \begin{bmatrix} \left(\mathbf{R}_i^j\right)^{\mathrm{T}} & -\left(\mathbf{R}_i^j\right)^{\mathrm{T}} \mathbf{d}_i^j \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \tag{2.3}$$

gegeben.

Die Greifhand und die Kamera werden als Punktmasse betrachtet und das Robotermodell um diese erweitert. Die Position und Rotation der Kamerabasis $(0_c x_c y_c z_c)$ und der Greifhand $(0_h x_h y_h z_h)$ in Bezug zum Basiskoordinatensystem $(0xyz)$ ergeben sich aus den homogenen Transformationen $\mathbf{H}_0^c$ und $\mathbf{H}_0^h$ in Abhängigkeit der Gelenkswinkel $\mathbf{q}$ zu

$$\mathbf{H}_0^c(\mathbf{q}) = \mathbf{H}_0^1\mathbf{H}_1^2\mathbf{H}_2^3\mathbf{H}_3^4\mathbf{H}_4^5\mathbf{H}_5^6\mathbf{H}_6^7\mathbf{H}_7^c, \tag{2.4a}$$

$$\mathbf{H}_0^h(\mathbf{q}) = \mathbf{H}_0^1\mathbf{H}_1^2\mathbf{H}_2^3\mathbf{H}_3^4\mathbf{H}_4^5\mathbf{H}_5^6\mathbf{H}_6^7\mathbf{H}_7^h. \tag{2.4b}$$

Für die Herleitung des dynamischen Modells werden noch die homogenen Transformationen

der Schwerpunkte der einzelnen Starrkörper des Roboters, der Greifhand und der Kamera

$$\mathbf{H}_0^{b1}(\mathbf{q}) = \mathbf{H}_0^1\mathbf{H}_1^{b1}, \tag{2.5a}$$
$$\mathbf{H}_0^{b2}(\mathbf{q}) = \mathbf{H}_0^1\mathbf{H}_1^2\mathbf{H}_2^{b2}, \tag{2.5b}$$
$$\mathbf{H}_0^{b3}(\mathbf{q}) = \mathbf{H}_0^1\mathbf{H}_1^2\mathbf{H}_2^3\mathbf{H}_3^{b3}, \tag{2.5c}$$
$$\mathbf{H}_0^{b4}(\mathbf{q}) = \mathbf{H}_0^1\mathbf{H}_1^2\mathbf{H}_2^3\mathbf{H}_3^4\mathbf{H}_4^{b4}, \tag{2.5d}$$
$$\mathbf{H}_0^{b5}(\mathbf{q}) = \mathbf{H}_0^1\mathbf{H}_1^2\mathbf{H}_2^3\mathbf{H}_3^4\mathbf{H}_4^5\mathbf{H}_5^{b5}, \tag{2.5e}$$
$$\mathbf{H}_0^{b6}(\mathbf{q}) = \mathbf{H}_0^1\mathbf{H}_1^2\mathbf{H}_2^3\mathbf{H}_3^4\mathbf{H}_4^5\mathbf{H}_5^6\mathbf{H}_6^{b6}, \tag{2.5f}$$
$$\mathbf{H}_0^{b7}(\mathbf{q}) = \mathbf{H}_0^1\mathbf{H}_1^2\mathbf{H}_2^3\mathbf{H}_3^4\mathbf{H}_4^5\mathbf{H}_5^6\mathbf{H}_6^7\mathbf{H}_7^{b7} \tag{2.5g}$$
$$\mathbf{H}_0^{bc}(\mathbf{q}) = \mathbf{H}_0^c\mathbf{H}_c^{bc} \tag{2.5h}$$
$$\mathbf{H}_0^{bh}(\mathbf{q}) = \mathbf{H}_0^h\mathbf{H}_h^{bh} \tag{2.5i}$$
$$\tag{2.5j}$$

benötigt. Die jeweiligen Rotationsmatrizen und Translationsvektoren für $\mathbf{H}_i^j$ in (2.4) und (2.5) sind in Tabelle A.1 im Anhang angegeben.

2.2.2 Differentielle Kinematik

Die differentielle Kinematik beschreibt den Zusammenhang zwischen der translatorischen Geschwindigkeit $\mathbf{v}_i, i \in \{b1,b2,\ldots,b7,bc,bh\}$ und der Winkelgeschwindigkeit $\boldsymbol{\omega}_i, i \in \{b1,b2,\ldots,b7,bc,bh\}$ der einzelnen Starrkörper des Roboters, der Kamera und der Hand in Abhängigkeit der Geschwindigkeiten der Robotergelenke $\dot{\mathbf{q}}$. Für eine einfachere Darstellung gilt für die hochgestellten Indizes in diesem Abschnitt $i \in \{b1,b2,\ldots,b7,bc,bh\}$.

Mit den Manipulator Jacobi-Matrizen $\mathbf{J}_0^i(\mathbf{q})$ werden die Winkelgeschwindigkeiten der Gelenke $\dot{\mathbf{q}}$ auf $\mathbf{v}_0^i$ und $\boldsymbol{\omega}_0^i$ abgebildet [15], entsprechend

$$\begin{bmatrix} \mathbf{v}_0^i \\ \boldsymbol{\omega}_0^i \end{bmatrix} = \begin{bmatrix} (\mathbf{J}_\mathbf{v})_0^i(\mathbf{q}) \\ (\mathbf{J}_\boldsymbol{\omega})_0^i(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}_0^i(\mathbf{q})\,\dot{\mathbf{q}}, \tag{2.6}$$

mit

$$(\mathbf{J}_\mathbf{v})_0^i(\mathbf{q}) = \frac{\partial \mathbf{d}_0^i(\mathbf{q})}{\partial \mathbf{q}}, \tag{2.7a}$$
$$(\mathbf{J}_\boldsymbol{\omega})_0^i(\mathbf{q}) = \frac{\partial \boldsymbol{\omega}_0^i}{\partial \dot{\mathbf{q}}}. \tag{2.7b}$$

Die Vektoren der Winkelgeschwindigkeiten $\boldsymbol{\omega}_0^i$ können ebenfalls mit Hilfe der schiefsymmetrischen Matrix [15]

$$\mathbf{S}\left(\boldsymbol{\omega}_0^i\right) = \dot{\mathbf{R}}_0^i\left(\mathbf{R}_0^i\right)^\mathrm{T} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.8}$$

gebildet werden.

2.2.3 Dynamisches Modell

Das dynamische Robotermodell wird mit Hilfe der Euler-Lagrangegleichungen

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} - \left(\frac{\partial \mathcal{L}}{\partial \mathbf{q}}\right)^{\mathrm{T}} = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \tag{2.9}$$

mit der Lagrange Funktion $\mathcal{L} = \mathcal{T} - \mathcal{V}$ hergeleitet, wobei mit $\mathcal{L}$ die Lagrange Funktion bezeichnet wird und $\boldsymbol{\tau}$ die generalisierten Kräfte bezeichnen. Durch $\boldsymbol{\tau}_{ext}$ werden externe und dissipative Kräfte berücksichtigt. Die kinetische Energie $\mathcal{T}$ und die potentielle Energie $\mathcal{V}$ aufgrund der Schwerkraft ergeben sich aus Summe der kinetischen und potentiellen Energien der einzelnen Starrkörper $i \in \{b1, b2, \ldots, b7, bc, bh\}$ des Roboters, der Kamera und der Greifhand im Basiskoordinatensystem, gemäß

$$\mathcal{T} = \frac{1}{2} \sum_{i \in \{b1,b2,\ldots,b7,bc,bh\}} \left[\left(\boldsymbol{\omega}_0^i\right)^{\mathrm{T}} \mathbf{R}_0^i \mathbf{I}_i \left(\mathbf{R}_0^i\right)^{\mathrm{T}} \boldsymbol{\omega}_0^i + m_i \left(\mathbf{v}_0^i\right)^{\mathrm{T}} \mathbf{v}_0^i \right], \tag{2.10a}$$

$$\mathcal{V} = \sum_{i \in \{b1,b2,\ldots,b7,bc,bh\}} m_i g \mathbf{e}_g^{\mathrm{T}} \mathbf{d}_0^i, \tag{2.10b}$$

mit dem Einheitsvektor der Schwerkraft $\mathbf{e}_g = [0 \quad 0 \quad 1]^{\mathrm{T}}$, den Massen m_i und den körperfesten Trägheitsmatrizen $\mathbf{I}_i = \mathrm{diag}\left([I_{i,xx}, I_{i,yy}, I_{i,zz}]\right)$ im Schwerpunkt der einzelnen Starrkörper. Die translatorischen Geschwindigkeiten $\mathbf{v}_i$ und die Winkelgeschwindigkeiten $\boldsymbol{\omega}_i$ sind durch die Beziehung (2.6) gegeben.

Die Bewegungsgleichungen eines Roboters werden üblicherweise in der Form

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \tag{2.11}$$

angegeben, mit der generalisierten Massenmatrix $\mathbf{D}(\mathbf{q})$, der Coriolismatrix $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$, dem Vektor der Potentialterme $\mathbf{g}(\mathbf{q})$, den Gelenksdrehmomenten $\boldsymbol{\tau}$ und den generalisierten externen Kräften $\boldsymbol{\tau}_{ext}$. Die Massenmatrix $\mathbf{D}(\mathbf{q})$ ergibt sich durch Einsetzen von (2.6) in (2.10a) zu

$$\mathcal{T} = \frac{1}{2}\dot{\mathbf{q}} \underbrace{\sum_{i \in \{b1,b2,\ldots,b7,bc,bh\}} \left[\left((\mathbf{J}_\omega)_0^i(\mathbf{q})\right)^{\mathrm{T}} \mathbf{R}_0^i \mathbf{I}_i \left(\mathbf{R}_0^i\right)^{\mathrm{T}} (\mathbf{J}_\omega)_0^i(\mathbf{q}) + m_i \left((\mathbf{J}_\mathbf{v})_0^i(\mathbf{q})\right)^{\mathrm{T}} (\mathbf{J}_\mathbf{v})_0^i(\mathbf{q}) \right]}_{\mathbf{D}(\mathbf{q})} \dot{\mathbf{q}}. \tag{2.12}$$

In der Coriolismatrix $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ sind die Zentrifugal- und Coriolisterme enthalten. Die Elemente der Coriolismatrix werden über

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})_{[k,j]} = \sum_{i=1}^{n} c_{ijk}(\mathbf{q})\dot{q}_i \tag{2.13}$$

berechnet. Dabei bezeichnen $c_{ijk}(\mathbf{q})$ die sog. Christoffel Symbole, die sich mit den (k,j)-ten Elemente der Massenmatrix $d_{kj}(\mathbf{q}) = \mathbf{D}(\mathbf{q})_{[k,j]}$ aus

$$c_{ijk}(\mathbf{q}) = \frac{1}{2}\left(\frac{\partial d_{kj}(\mathbf{q})}{\partial q_i} + \frac{\partial d_{ki}(\mathbf{q})}{\partial q_j} + \frac{\partial d_{ij}(\mathbf{q})}{\partial q_k}\right) \tag{2.14}$$

ergeben. Bei dem KUKA LBR iiwa 14 R820 Roboter handelt es sich um einen Roboter mit flexiblen Gelenken. Für solche Roboter ist das Modell (2.11) nicht ausreichend und wird auf Basis von [12] erweitert. Ein flexibles Gelenk wird mit einer Drehfeder zwischen der Motorachse und der Gelenksachse mit den Federkonstanten $\mathbf{K}$ modelliert. Das wirkende Drehmoment an der Gelenksachse ist proportional zur Differenz des Winkels der Motorachse $\boldsymbol{\theta}$ und des Gelenkswinkels $\mathbf{q}$

$$\boldsymbol{\tau} = \mathbf{K}\left(\boldsymbol{\theta} - \mathbf{q}\right). \tag{2.15}$$

Die Dynamik der Motorachse des Roboters wird mit der Matrix der Rotorträgheitsmomente $\mathbf{B}$ und den Drehmomenten der Motoren $\boldsymbol{\tau}_m$ über

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \mathbf{K}\left(\boldsymbol{\theta} - \mathbf{q}\right) = \boldsymbol{\tau}_m \tag{2.16}$$

beschrieben. Einsetzen von (2.15) in (2.11) ergibt zusammen mit der Rotordynamik (2.16) das Modell eines Roboters mit flexiblen Gelenken

$$\mathbf{D}\left(\mathbf{q}\right)\ddot{\mathbf{q}} + \mathbf{C}\left(\mathbf{q},\dot{\mathbf{q}}\right)\dot{\mathbf{q}} + \mathbf{g}\left(\mathbf{q}\right) = \mathbf{K}\left(\boldsymbol{\theta} - \mathbf{q}\right) + \boldsymbol{\tau}_{ext}, \tag{2.17a}$$

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \mathbf{K}\left(\boldsymbol{\theta} - \mathbf{q}\right) = \boldsymbol{\tau}_m. \tag{2.17b}$$

Auf Basis der singulären Störtheorie kann gezeigt werden, dass es sich um ein schnelles und ein langsames Teilsystem handelt. Zusätzlich zu den Zustandsgrößen $\mathbf{q}$ und $\dot{\mathbf{q}}$ stehen beim verwendeten Roboter auch die Drehmomente der jeweiligen Gelenke als Messgrößen $\boldsymbol{\tau}_s$ zur Verfügung. Aus diesen Messgrößen wird über ein Differenzierungsfilter $\dot{\boldsymbol{\tau}}_s$ gebildet. Durch das PD-Zustandsregelgesetz

$$\boldsymbol{\tau}_m = \boldsymbol{\tau}_d - \mathbf{K}_\tau\left(\boldsymbol{\tau}_s - \boldsymbol{\tau}_d\right) - \epsilon\mathbf{D}_\tau\dot{\boldsymbol{\tau}}_s \tag{2.18}$$

ergibt sich schließlich der geschlossene Kreis zu [12]

$$\left(\mathbf{D}\left(\mathbf{q}\right) + \left(\mathbf{E} + \mathbf{K}_\tau\right)^{-1}\mathbf{B}\right)\ddot{\mathbf{q}} + \mathbf{C}\left(\mathbf{q},\dot{\mathbf{q}}\right)\dot{\mathbf{q}} + \mathbf{g}\left(\mathbf{q}\right) = \boldsymbol{\tau}_d. \tag{2.19}$$

Dabei bezeichnet $\mathbf{E}$ die Einheitsmatrix, $\mathbf{K}_\tau$ und $\mathbf{D}_\tau$ sind Verstärkungsmatrizen des Reglers der singulären Störtheorie und $\boldsymbol{\tau}_d$ das gewünschte Drehmoment aus einem Regler für das Starrkörpermodell.

2.3 Greifhand Schunk SDH2

Im folgenden Abschnitt wird die Vorwärtskinematik der Greifhand hergeleitet. Für den Greifpunkt an der Hand wird ein zusätzlicher Punkt G definiert, siehe dazu Abbildung 2.7. Wird eine Linie zwischen den Mittelpunkten C_1 und C_3 der Rundung der Greiffläche gezeichnet, liegt der Punkt G auf dem Mittelpunkt dieser Verbindungslinie. Für eine bessere Planung des dynamischen Greifens wird zusätzlich zum Punkt G ein Greifzustand für die Distanz zwischen den Greifflächen der Fingerspitzen und einer für die Orientierung der Tangente der Greiffläche bei geschlossenen Fingern eingeführt. Durch die Einführung dieser Greifzustände wird eine intuitivere Planung des Greifvorgangs ermöglicht. In Abschnitt 2.3.2 wird die inverse Kinematik für die Berechnung der Zustandsgrößen der Greifhand aus den Greifzuständen hergeleitet. Weiters wird in Abschnitt 2.3.3 ein zulässiges Gebiet für die Lösbarkeit der inversen Kinematik definiert.

2.3.1 Vorwärtskinematik

Wie bereits erwähnt, wird der Finger 2 für das Greifen des Objekts nicht benötigt. Die Zustandsgrößen des Fingers 2 werden auf die konstanten Werte $q_{h4} = q_{h4,0}$ und $q_{h5} = q_{h5,0}$ gesetzt und somit wird der Greifvorgang durch den Finger 2 nicht behindert. Da der Greifvorgang mit Finger 1 und 3 ausgeführt wird, muss die Drehung der beiden Finger um die jeweilige Hochachse konstant auf den Wert $q_{h1} = q_{h1,0} = 90°$ gesetzt werden, damit die Greifflächen der beiden Finger parallel zueinander sind.

Die Vorwärtskinematik $\mathbf{p}_0^G(\mathbf{q}_h)$ beschreibt die Position des Punktes G in Abhängigkeit der Gelenkskoordinaten $\mathbf{q}_h$. Zur besseren Lesbarkeit wird im Weiteren auf die Argumente $\mathbf{q}_h$ verzichtet. In diesem Abschnitt bezeichnet der Vektor $\mathbf{p}_j^k$ eine Translation und die Matrix $\mathbf{R}_j^k$ eine Rotation zwischen zwei Koordinatensystemen $(0_j x_j y_j z_j)$ und $(0_k x_k y_k z_k)$.

Die Translationen zwischen den Koordinatensystemen der Hand $(0_{hi} x_{hi} y_{hi} z_{hi}), i \in \{1,2,\ldots,6\}$ mit der Basis $(0_h x_h y_h z_h)$ ergeben sich mit den in Abbildung 2.3 dargestellten Parametern zu

$$\mathbf{p}_h^{h1} = [l_1, -b_1, h_1]^{\mathrm{T}}, \qquad \mathbf{p}_{h1}^{h2} = [0, 0, h_2]^{\mathrm{T}}, \qquad \mathbf{p}_{h2}^{C1} = [0, s_C \sin\theta, s_C \cos\theta]^{\mathrm{T}}, \tag{2.20a}$$
$$\mathbf{p}_h^{h3} = [-l_2, 0, h_1]^{\mathrm{T}}, \qquad \mathbf{p}_{h3}^{h4} = \mathbf{p}_{h1}^{h2}, \qquad \mathbf{p}_{h4}^{C2} = \mathbf{p}_{h2}^{C1}, \tag{2.20b}$$
$$\mathbf{p}_h^{h5} = [l_1, b_1, h_1]^{\mathrm{T}}, \qquad \mathbf{p}_{h5}^{h6} = \mathbf{p}_{h1}^{h2}, \qquad \mathbf{p}_{h6}^{C3} = \mathbf{p}_{h2}^{C1}. \tag{2.20c}$$

Die Freiheitsgrade der Hand $q_{hi}, i \in \{1,2,\ldots,7\}$ sind in Abbildung 2.2 und 2.7 eingezeichnet, daraus ergeben sich die Rotationsmatrizen

$$\mathbf{R}_h^{h1} = \mathbf{R}_{z_h,-\frac{\pi}{2}-q_{h1}} \mathbf{R}_{x_{h1},q_{h2}}\Big|_{q_{h1}=q_{h1,0}} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -\cos(q_{h2}) & \sin(q_{h2}) \\ 0 & \sin(q_{h2}) & \cos(q_{h2}) \end{bmatrix}, \tag{2.21a}$$

$$\mathbf{R}_{h1}^{h2} = \mathbf{R}_{x_{h2},q_{h3}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(q_{h3}) & -\sin(q_{h3}) \\ 0 & \sin(q_{h3}) & \cos(q_{h3}) \end{bmatrix}, \tag{2.21b}$$

$$\mathbf{R}_h^{h3} = \mathbf{R}_{z_h,\frac{\pi}{2}} \mathbf{R}_{x_{h3},q_{h4}}\Big|_{q_{h4}=q_{h4,0}} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{2.21c}$$

$$\mathbf{R}_{h3}^{h4} = \mathbf{R}_{x_{h4},q_{h5}}\big|_{q_{h5}=q_{h5,0}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix}, \tag{2.21d}$$

$$\mathbf{R}_h^{h5} = \mathbf{R}_{z_h,-\frac{\pi}{2}+q_{h1}} \mathbf{R}_{x_{h5},q_{h6}}\Big|_{q_{h1}=q_{h1,0}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(q_{h6}) & -\sin(q_{h6}) \\ 0 & \sin(q_{h6}) & \cos(q_{h6}) \end{bmatrix}, \tag{2.21e}$$

$$\mathbf{R}_{h5}^{h6} = \mathbf{R}_{x_{h6},q_{h7}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(q_{h7}) & -\sin(q_{h7}) \\ 0 & \sin(q_{h7}) & \cos(q_{h7}) \end{bmatrix}, \tag{2.21f}$$

wobei für q_{h1}, q_{h4} und q_{h5} die Initialwerte eingesetzt werden. Die Position der Koordina-

tensysteme der Fingerspitzen ist gegeben durch

$$\mathbf{p}_h^{h2} = \mathbf{p}_h^{h1} + \mathbf{R}_h^{h1}\mathbf{p}_{h1}^{h2} = [l_1, -b_1 + h_2 \sin(q_{h2}), h_1 + h_2 \cos(q_{h2})]^{\mathrm{T}}, \tag{2.22a}$$

$$\mathbf{p}_h^{h6} = \mathbf{p}_h^{h5} + \mathbf{R}_h^{h5}\mathbf{p}_{h5}^{h6} = [l_1, b_1 - h_2 \sin(q_{h6}), h_1 + h_2 \cos(q_{h6})]^{\mathrm{T}} \tag{2.22b}$$

und die Position der Zentren der Rundungen der Fingerspitzen durch

$$\mathbf{p}_h^{C1} = \mathbf{p}_h^{h2} + \mathbf{R}_h^{h1}\mathbf{R}_{h1}^{h2}\mathbf{p}_{h2}^{C1} \tag{2.23a}$$

$$= \begin{bmatrix} l_1 \\ -b_1 + h_2 \sin(q_{h2}) + s_C \sin(q_{h2} + q_{h3} - \theta) \\ h_1 + h_2 \cos(q_{h2}) + s_C \cos(q_{h2} + q_{h3} - \theta) \end{bmatrix} = \begin{bmatrix} x_{C1} \\ y_{C1} \\ z_{C1} \end{bmatrix}, \tag{2.23b}$$

$$\mathbf{p}_h^{C3} = \mathbf{p}_h^{h6} + \mathbf{R}_h^{h5}\mathbf{R}_{h5}^{h6}\mathbf{p}_{h6}^{C3} \tag{2.24a}$$

$$= \begin{bmatrix} l_1 \\ b_1 - h_2 \sin(q_{h6}) - s_C \sin(q_{h6} + q_{h7} - \theta) \\ h_1 + h_2 \cos(q_{h6}) + s_C \cos(q_{h6} + q_{h7} - \theta) \end{bmatrix} = \begin{bmatrix} x_{C3} \\ y_{C3} \\ z_{C3} \end{bmatrix}. \tag{2.24b}$$

Die Position des Greifpunktes G berechnet sich aus

$$\mathbf{p}_h^{G} = \mathbf{p}_h^{C1} + \frac{\mathbf{p}_h^{C3} - \mathbf{p}_h^{C1}}{2}. \tag{2.25}$$

2.3.2 Inverse Kinematik

Für die Planung eines Griffs werden vier neue Zustandsgrößen $\mathbf{q}_G = [d_G, \varphi_G, y_G, z_G]^{\mathrm{T}}$ eingeführt, durch welche der Greifzustand beschrieben wird. Sie sind in Abbildung 2.7 dargestellt. Dabei bezeichnen y_G und z_G die y- und z-Komponenten des Greifpunktes G in Bezug auf das Koordinatensystem $(0_h x_h y_h z_h)$. φ_G beschreibt den Greifwinkel und d_G definiert wie weit die Finger geöffnet sind. In diesem Abschnitt wird die inverse Kinematik hergeleitet. Damit lassen sich die Zustandsgrößen q_{h2}, q_{h3}, q_{h6} und q_{h7} in Abhängigkeit der Zustandsgrößen $\mathbf{q}_G$ berechnen. Zur besseren Lesbarkeit wird im Weiteren auf die Argumente $\mathbf{q}_G$ verzichtet.

Die y- und z-Komponenten der Mittelpunkte der Rundungen der Fingerspitzen C_1 und C_3 ergeben sich mit den Greifzuständen $\mathbf{q}_G$ zu (siehe Abbildung 2.3 und Abbildung 2.7)

$$y_{C1} = y_G - \left(r_{T2} + \frac{d_G}{2}\right)\cos(\varphi_G), \tag{2.26}$$

$$z_{C1} = z_G - \left(r_{T2} + \frac{d_G}{2}\right)\sin(\varphi_G), \tag{2.27}$$

$$y_{C3} = y_G + \left(r_{T2} + \frac{d_G}{2}\right)\cos(\varphi_G), \tag{2.28}$$

$$z_{C3} = z_G + \left(r_{T2} + \frac{d_G}{2}\right)\sin(\varphi_G). \tag{2.29}$$

Um einen Ausdruck für q_{h2} bzw. q_{h6} zu erhalten, werden jeweils die y- und z-Komponenten

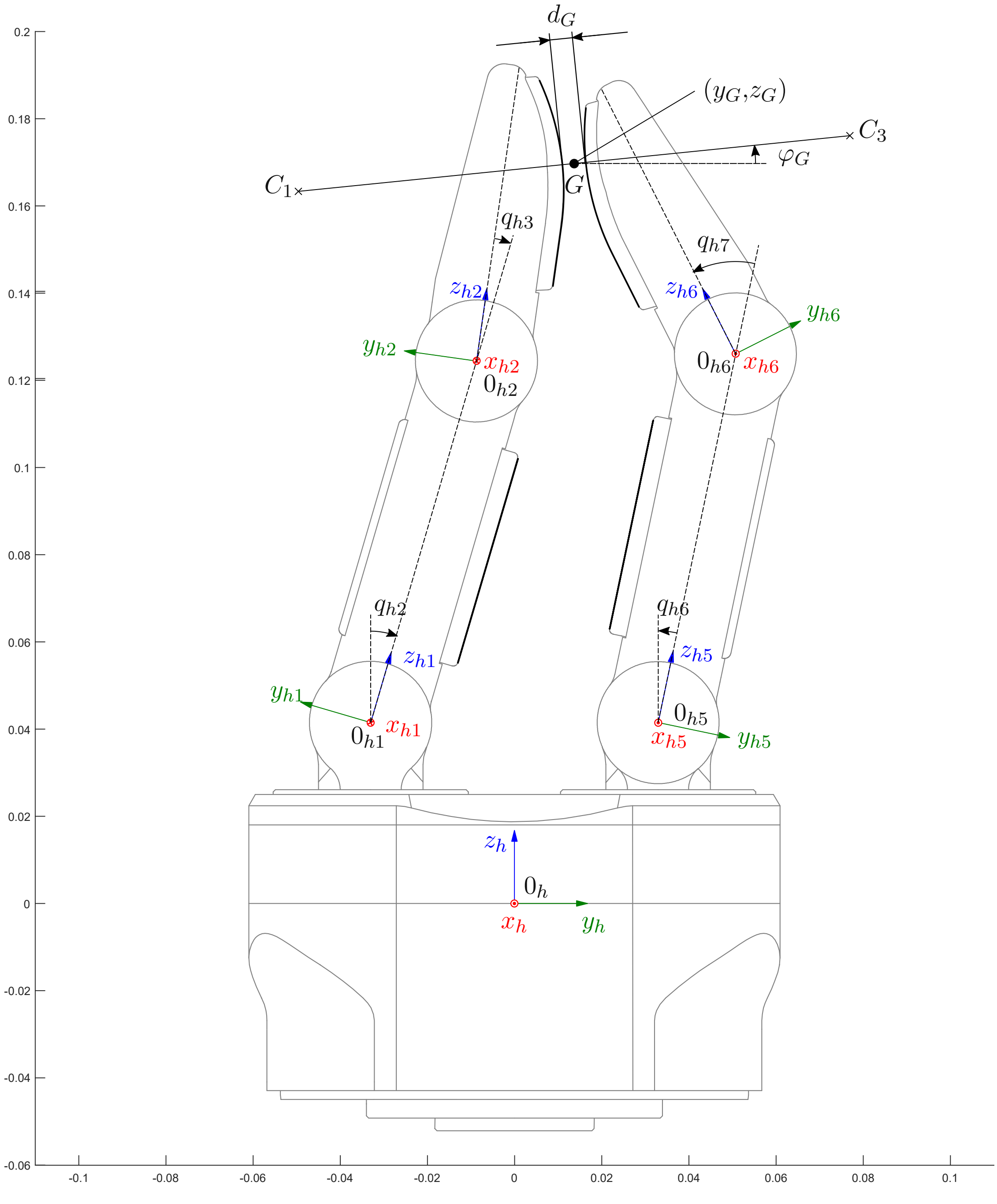


Abbildung 2.7: Seitenansicht der Greifhand mit den eingezeichneten Zustandsgrößen $\mathbf{q}_h$ und dem Greifzustand $\mathbf{q}_G$.

von (2.23b) und (2.24b) quadriert, addiert und alle Terme auf die linke Seite gebracht, entsprechend

$$\begin{aligned}2\,(h_1 - z_{C1})\, h_2 \cos(q_{h2}) - 2\,(y_{C1} + b_1)\, h_2 \sin(q_{h2}) + \\ b_1^2 + 2b_1 y_{C1} + h_1^2 - 2h_1 z_{C1} + h_2^2 + y_{C1}^2 + z_{C1}^2 - s_C^2 = 0,\end{aligned} \tag{2.30a}$$

$$\begin{aligned}2\,(h_1 - z_{C3})\, h_2 \cos(q_{h6}) - 2\,(b_1 - y_{C3})\, h_2 \sin(q_{h6}) + \\ b_1^2 - 2b_1 y_{C3} + h_1^2 - 2h_1 z_{C3} + h_2^2 + y_{C3}^2 + z_{C3}^2 - s_C^2 = 0.\end{aligned} \tag{2.30b}$$

Zusätzlich werden folgende Substitutionen

$$t_i = \tan\left(\frac{q_{hi}}{2}\right), \quad \sin(q_{hi}) = \frac{2t_i}{1+t_i^2}, \quad \cos(q_{hi}) = \frac{1-t_i^2}{1+t_i^2}, \quad i \in \{2{,}6\} \tag{2.31}$$

und

$$e_1 = -2\,(y_{C1} + b_1)\, h_2, \tag{2.32a}$$
$$e_2 = 2\,(h_1 - z_{C1})\, h_2, \tag{2.32b}$$
$$e_3 = b_1^2 + 2b_1 y_{C1} + h_1^2 - 2h_1 z_{C1} + h_2^2 - s_C^2 + y_{C1}^2 + z_{C1}^2, \tag{2.32c}$$
$$e_4 = -2\,(b_1 - y_{C3})\, h_2, \tag{2.32d}$$
$$e_5 = 2\,(h_1 - z_{C3})\, h_2, \tag{2.32e}$$
$$e_6 = b_1^2 - 2b_1 y_{C3} + h_1^2 - 2h_1 z_{C3} + h_2^2 - s_C^2 + y_{C3}^2 + z_{C3}^2 \tag{2.32f}$$

eingeführt. Einsetzen von (2.31) und (2.32) in (2.30) führt zu den quadratischen Gleichungen

$$e_1 \frac{2t_2}{1+t_2^2} + e_2 \frac{1-t_2^2}{1+t_2^2} + e_3 = 0, \tag{2.33a}$$
$$e_4 \frac{2t_6}{1+t_6^2} + e_5 \frac{1-t_6^2}{1+t_6^2} + e_6 = 0. \tag{2.33b}$$

Die Lösung dieser quadratischen Gleichungen ist

$$t_{2_{1,2}} = \frac{-e_1 \pm \sqrt{e_1^2 + e_2^2 - e_3^2}}{-e_2 + e_3}, \tag{2.34a}$$
$$t_{6_{1,2}} = \frac{-e_4 \pm \sqrt{e_4^2 + e_5^2 - e_6^2}}{-e_5 + e_6}. \tag{2.34b}$$

Für den Tangens von q_{h2} und q_{h6} existieren zwei Lösungen. Um einen möglichst großen Winkel zwischen den proximalen Teilen der Finger zu erreichen, wird die größere Lösung der quadratischen Gleichung für die Berechnung der Winkel q_{h2} und q_{h6} verwendet, gemäß

$$q_{h2} = -2\arctan\left(\frac{-e_1 + \sqrt{e_1^2 + e_2^2 - e_3^2}}{e_2 - e_3}\right), \tag{2.35a}$$
$$q_{h6} = -2\arctan\left(\frac{-e_4 + \sqrt{e_4^2 + e_5^2 - e_6^2}}{e_5 - e_6}\right). \tag{2.35b}$$

Einsetzen von (2.35) in die y-Komponente von (2.23b) und (2.24b) ergibt

$$q_{h3} = -q_{h2} + \theta - \arcsin\left(\frac{\sin\left(q_{h2}\right)h_2 - b_1 - y_{C1}}{s_C}\right), \tag{2.36a}$$

$$q_{h7} = -q_{h6} + \theta - \arcsin\left(\frac{\sin\left(q_{h6}\right)h_2 - b_1 + y_{C3}}{s_C}\right). \tag{2.36b}$$

Zusammengefasst in einem Vektor ergibt sich die inverse Kinematik zu

$$\mathbf{q}_h\left(\mathbf{q}_G\right) = \begin{bmatrix} q_{h1,0} \\ -2\arctan\left(\frac{-e_1+\sqrt{e_1^2+e_2^2-e_3^2}}{e_2-e_3}\right) \\ -q_{h2} + \theta - \arcsin\left(\frac{\sin(q_{h2})h_2-b_1-y_{C1}}{s_C}\right) \\ q_{h4,0} \\ q_{h5,0} \\ -2\arctan\left(\frac{-e_4+\sqrt{e_4^2+e_5^2-e_6^2}}{e_5-e_6}\right) \\ -q_{h6} + \theta - \arcsin\left(\frac{\sin(q_{h6})h_2-b_1+y_{C3}}{s_C}\right) \end{bmatrix}, \tag{2.37}$$

wobei $q_{h1,0}$, $q_{h4,0}$ und $q_{h5,0}$ als konstant angenommen werden.

2.3.3 Zulässiges Gebiet

Nicht für alle Werte von $\mathbf{q}_G$ existiert eine Lösung der Rückwärtskinematik. Zwei Begrenzungen ergeben sich aus der Lösung der quadratischen Gleichung (2.34). Um ein reelles Ergebnis zu erhalten, müssen die Terme unter den Wurzeln die Bedingungen

$$-e_1^2 - e_2^2 + e_3^2 \leq 0, \qquad -e_4^2 - e_5^2 + e_6^2 \leq 0 \tag{2.38}$$

erfüllen. Die maximale Höhe des Greifpunktes z_G, bezogen auf die Basis der Hand $(0_h x_h y_h z_h)$, wird durch die oberen Enden der Greifflächen der Fingerspitzen limitiert. Es werden, wie in Abbildung 2.8a eingezeichnet, an den oberen Enden der Greifflächen die Punkte T_1 und T_3 mit den Ortsvektoren $\mathbf{p}_{T1}$ und $\mathbf{p}_{T3}$ gemäß

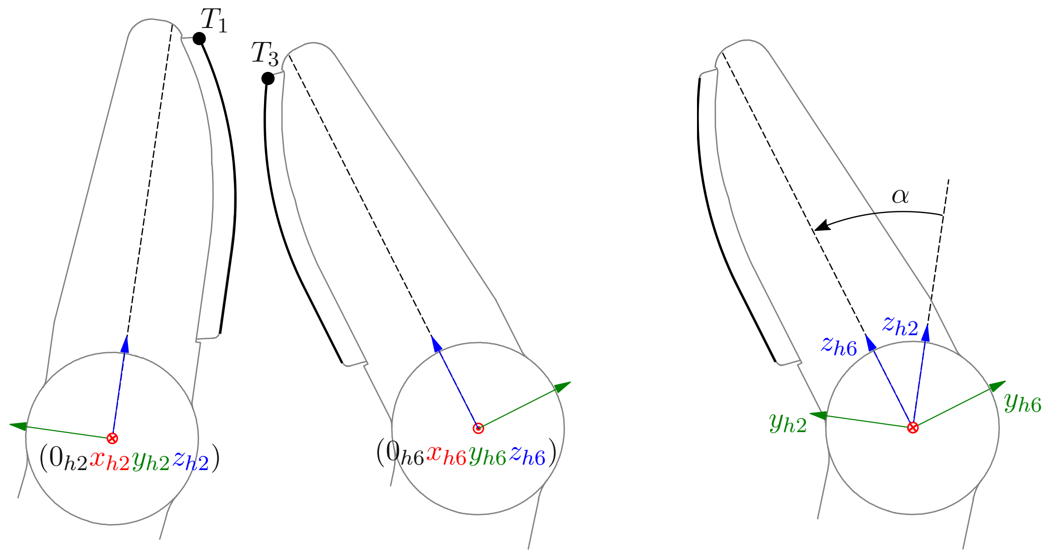
$$\mathbf{p}_{T1} = \begin{bmatrix} l_1 \\ -b_1 + h_2\sin\left(q_{h2}\right) + s_T\sin\left(q_{h2}+q_{h3}+\eta\right) \\ h_1 + h_2\cos\left(q_{h2}\right) + s_T\cos\left(q_{h2}+q_{h3}+\eta\right) \end{bmatrix}, \tag{2.39a}$$

$$\mathbf{p}_{T3} = \begin{bmatrix} l_1 \\ b_1 - h_2\sin\left(q_{h6}\right) - s_T\sin\left(q_{h6}+q_{h7}+\eta\right) \\ h_1 + h_2\cos\left(q_{h6}\right) + s_T\cos\left(q_{h6}+q_{h7}+\eta\right) \end{bmatrix} \tag{2.39b}$$

eingeführt. Für einen erfolgreichen Griff muss z_G gleich oder kleiner sein als die z-Komponenten der Vektoren $\mathbf{p}_{T1}$ und $\mathbf{p}_{T3}$. Daraus ergeben sich die Ungleichungsbeschränkungen

$$-h_1 - h_2\cos\left(q_{h2}\right) - s_T\cos\left(q_{h2}+q_{h3}+\eta\right) + z_G \leq 0, \tag{2.40}$$

$$-h_1 - h_2\cos\left(q_{h6}\right) - s_T\cos\left(q_{h6}+q_{h7}+\eta\right) + z_G \leq 0. \tag{2.41}$$



(a) Seitenansicht mit den Punkten T_1 und T_3

(b) Winkel α

Abbildung 2.8: Darstellung der Punkte T_1 und T_3 und des Winkels α.

Die letzte Gleichung für das zulässige Gebiet ergibt sich aus dem Winkel zwischen den beiden ebenen Greifflächen der Fingerspitze. Wird gedanklich das Koordinatensystem $(0_{h2}x_{h2}y_{h2}z_{h2})$ in das Koordinatensystem $(0_{h6}x_{h6}y_{h6}z_{h6})$ verschoben, siehe Abbildung 2.8b, dann muss der Winkel $\alpha \geq 0$ sein. Zusätzlich wird für α ein Maximalwert von 90° festgelegt, womit gilt $\alpha \in [0{,}90°]$. Aus der Transformation der Einheitsvektoren $\mathbf{e}_z$ der beiden Koordinatensysteme $(0_{h2}x_{h2}y_{h2}z_{h2})$ und $(0_{h6}x_{h6}y_{h6}z_{h6})$ in die Basis der Hand ergeben sich die Vektoren $\mathbf{e}_{z,h2}$ und $\mathbf{e}_{z,h6}$ zu

$$\mathbf{e}_{z,h2} = \mathbf{R}_h^{h1}\mathbf{R}_{h1}^{h2}\mathbf{e}_z, \tag{2.42}$$

$$\mathbf{e}_{z,h6} = \mathbf{R}_h^{h5}\mathbf{R}_{h5}^{h6}\mathbf{e}_z. \tag{2.43}$$

Aus Bildung der Norm des Kreuzproduktes zwischen den beiden Vektoren ergibt sich

$$\left\|\mathbf{R}_h^{h1}\mathbf{R}_{h1}^{h2}\mathbf{e}_z \times \mathbf{R}_h^{h5}\mathbf{R}_{h5}^{h6}\mathbf{e}_z\right\|_2 = \sin\left(q_{h2} + q_{h3} + q_{h6} + q_{h7}\right) = \sin\alpha \geq 0. \tag{2.44}$$

Damit beschreibt die Ungleichung

$$-\sin\left(q_{h2} + q_{h3} + q_{h6} + q_{h7}\right) \leq 0 \tag{2.45}$$

eine weitere Beschränkung für das zulässige Gebiet. Zusammengefasst wird das zulässige Gebiet durch die Ungleichungen

$$\mathbf{h}_G\left(\mathbf{q}_h,\mathbf{q}_G\right) = \begin{bmatrix} -e_1^2 - e_2^2 + e_3^2 \\ -e_4^2 - e_5^2 + e_6^2 \\ -h_1 - h_2\cos\left(q_{h2}\right) - s_T\cos\left(q_{h2} + q_{h3} + \eta\right) + z_G \\ -h_1 - h_2\cos\left(q_{h6}\right) - s_T\cos\left(q_{h6} + q_{h7} + \eta\right) + z_G \\ -\sin\left(q_{h2} + q_{h3} + q_{h6} + q_{h7}\right) \end{bmatrix} \leq \mathbf{0} \tag{2.46}$$

definiert.

3 Objekterkennung

Dieses Kapitel beschreibt die kamerabasierte Schätzung der Pose des Greifobjekts. In Abschnitt 3.1 wird die Bestimmung der Pose anhand eines Kamerabildes und mit Hilfe von ArUco-Markern erläutert. Diese werden geeignet an bekannten Positionen des zu greifenden Objekts angebracht. Um die geschätzte Pose zu glätten und die Roboter-Bewegung zu berücksichtigen, wird im zweiten Teil ein Kalman-Filter entworfen.

3.1 Kamerabasierte Posenschätzung

Als Modell für die Kamera wird das Pinhole Kamera Modell in Abbildung 3.1 verwendet [16]. Dieses Modell beschreibt wie ein Punkt $\tilde{\mathbf{P}}$ mit den homogenen Koordinaten $\tilde{\mathbf{P}} = [x_i, y_i, z_i, 1]^{\mathrm{T}}$ in einem beliebigen Koordinatensystem $(0_i x_i y_i z_i)$ auf den Punkt $\tilde{\mathbf{p}}$ mit den homogenen Koordinaten $\tilde{\mathbf{p}} = [u, v, 1]^{\mathrm{T}}$ in der Bildebene $(0uv)$ abgebildet wird, gemäß der homogenen Transformation

$$s\tilde{\mathbf{p}} = \mathbf{K}\left[\mathbf{R}_c^i | \mathbf{t}_c^i\right]\tilde{\mathbf{P}}. \tag{3.1}$$

Dabei ist $\mathbf{K}$ die Kamera-Matrix

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

mit den intrinsischen Parametern der Kamera f_x, f_y, c_x und c_y, welche die Koordinaten im Kamerakoordinatensystem $(0_c x_c y_c z_c)$ in die Bildebene projiziert. Die Werte der intrinsischen Parameter sind in Tabelle A.4 im Anhang angegeben. Die Matrix $[\mathbf{R}_c^i | \mathbf{t}_c^i]$ bildet sich aus der Rotation $\mathbf{R}_c^i$ und Translation $\mathbf{t}_c^i$ zwischen dem Kamerakoordinatensystem $(0_c x_c y_c z_c)$ und dem Koordinatensystem $(0_i x_i y_i z_i)$ zu

$$\left[\mathbf{R}_c^i | \mathbf{t}_c^i\right] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \tag{3.3}$$

Der Skalierungsfaktor s errechnet sich aus dem Skalarprodukts der homogenen Koordinaten $\tilde{\mathbf{P}}$ und der letzten Zeile von (3.3) zu

$$s = r_{31}x_i + r_{32}y_i + r_{33}z_i + t_z. \tag{3.4}$$

Schließlich ergibt sich die 3D-to-2D Projektion zu

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{s}\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\left[\mathbf{R}_c^i | \mathbf{t}_c^i\right]\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}. \tag{3.5}$$

3 Objekterkennung

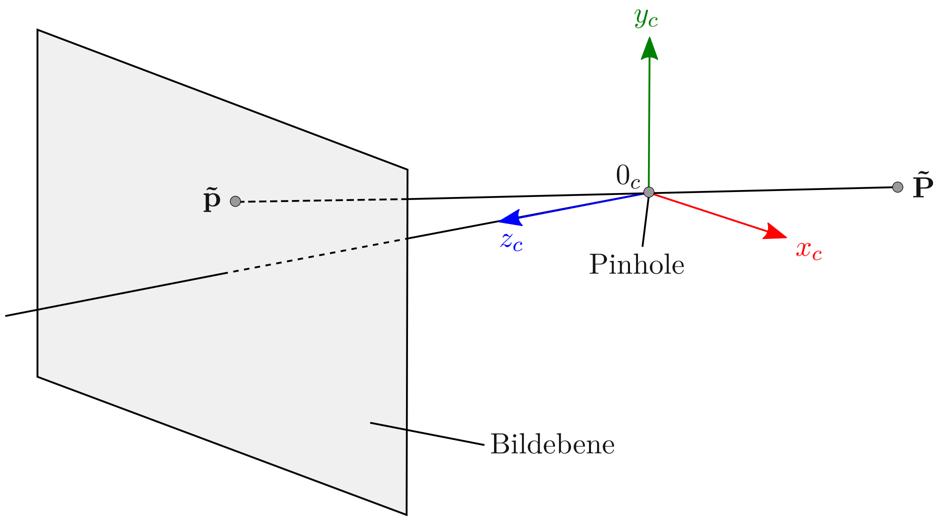


Abbildung 3.1: Pinhole Kamera Modell.

Die Werte u und v in (3.5) sind die Pixel-Koordinaten des Punktes $\tilde{\mathbf{P}}$ in der Bildebene. Ziel ist es, aus einem Kamerabild die Pose $[\mathbf{R}_c^i|\mathbf{t}_c^i]$ zu bestimmen. Für diese Aufgabe existieren unterschiedliche Implementierungen des Perspective-n-Point Algorithmus, welche $n \geq 3$ 3D-to-2D Projektionen entsprechend (3.5) benötigen. In dieser Arbeit wird der Efficient Perspective-n-Point, kurz EPnP, Algorithmus verwendet [2], welcher für $n \geq 4$ 3D-to-2D Projektionen entsprechend (3.5) angewendet werden kann.

Die Bestimmung der Pose des Greifobjekts $[\mathbf{R}_c^o|\mathbf{t}_c^o]$ erfolgt mit Hilfe von ArUco-Markern, siehe Abbildung 3.2, die an definierten Positionen auf dem Greifobjekt angebracht sind. Somit sind die Positionen aller Eckpunkte der einzelnen ArUco-Marker definiert. Zusätzlich ist jedem Marker durch die unterschiedliche Anordnung der weißen Quadrate im Inneren eine eindeutige ID zugewiesen. Damit können für jeden der vier Eckpunkte von n sichtbaren ArUco-Markern auf einem Bild die Punktepaare $(\tilde{\mathbf{p}}_i,\tilde{\mathbf{P}}_i), i \in \{1,\ldots,4n\}$ erstellt werden. Weiters werden die Punkte aus diesen Punktepaaren zu den Matrizen $\bar{\mathbf{p}} = [\tilde{\mathbf{p}}_1,\ldots,\tilde{\mathbf{p}}_{4n}]^\mathrm{T}$ und $\bar{\mathbf{P}} = \left[\tilde{\mathbf{P}}_1,\ldots,\tilde{\mathbf{P}}_{4n}\right]^\mathrm{T}$ zusammengefasst. Die Objektpose in Bezug auf das Kamerakoordinatensystem berechnet sich schließlich über den EPnP Algorithmus zu

$$[\mathbf{R}_c^o|\mathbf{t}_c^o] = \mathrm{EPnP}\left(\bar{\mathbf{p}},\bar{\mathbf{P}}\right). \tag{3.6}$$

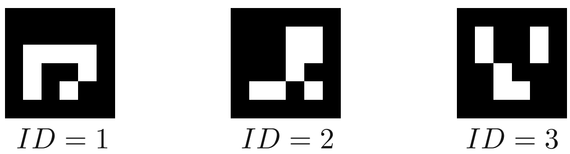


Abbildung 3.2: ArUco Marker mit der ID 1 bis 3.

3.2 Extended Kalman-Filter

Machine Vision Anwendungen sind sehr rechenintensiv. Das Auslesen des Bildes, identifizieren der ArUco-Marker und die Berechnung der Pose benötigt etliche Millisekunden. Da sich die Kamera am Endeffektor befindet, verändert sich die geschätzte Pose zwischen zwei Kamerabildern durch die Bewegung des Roboters. Während der Roboterbewegung wird, aufbauend auf der geschätzten Pose, zyklisch ein Trajektorienupdate des Roboters nach der in Kapitel 4.3 angeführten Methode ausgeführt. Der Startzeitpunkt des Trajektorienupdates ist dabei unabhängig von der Posenschätzung. Zusätzlich ist die Zyklusdauer für die einzelnen Trajektorienupdates nicht konstant. Somit ist nicht definiert, zu welchen Zeitpunkten die geschätzte Pose zur Berechnung des Trajektorienupdates verwendet wird. Um die Roboterbewegung zwischen zwei Kamerabildern zu berücksichtigen und damit die Posenschätzung für das Trajektorienupdate zu verbessern, wird ein Extended Kalman-Filter entworfen. Die Zustandsschätzung erfolgt anhand der Roboterbewegung und mit der berechneten Pose aus dem Kamerabild wird ein Messupdate durchgeführt.

Für den Entwurf des Extended Kalman-Filters wird zunächst ein nichtlineares zeitkontinuierliches Systemmodell für die Rotation und Distanz zwischen Objekt und Kamera hergeleitet. Dieses zeitkontinuierliche Modell wird im nächsten Schritt über das Euler-Verfahren in ein nichtlineares Abtastsystem überführt, auf dessen Basis schließlich das Extended Kalman-Filter entworfen wird. Die Herleitung der Systemmodelle und der Entwurf des Extended Kalman-Filters wird auf Basis von [17] durchgeführt.

Die Rotation zwischen zwei Koordinatensystemen $(0_i x_i y_i z_i)$ und $(0_j x_j y_j z_j)$ lässt sich in Form eines Einheitsquaternions mit

$$\mathbf{r} = \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} r_0 \\ \mathbf{r}_{1:3} \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) \\ \mathbf{n}^{\mathrm{T}} \sin\left(\frac{\alpha}{2}\right) \end{bmatrix}, \quad \|\mathbf{r}\|_2 = 1 \tag{3.7}$$

beschreiben [18]. Dabei bezeichnet α die Drehung um die Achse $\mathbf{n}$. Durch das Einsetzen von $(\alpha = \psi, \mathbf{n} = \mathbf{e}_x)$, $(\alpha = \theta, \mathbf{n} = \mathbf{e}_y)$ und $(\alpha = \phi, \mathbf{n} = \mathbf{e}_z)$ ergeben sich die elementaren Rotationsmatrizen aus (2.1). Die Hintereinanderausführung zweier Rotationen wird über das Quaternionprodukt $\otimes$ der entsprechenden Quaternionen beschrieben. Dabei ist das Quaternionprodukt zweier Quaterionen $\mathbf{r}$ und $\bar{\mathbf{r}}$ über

$$\mathbf{r} \otimes \bar{\mathbf{r}} = \begin{bmatrix} r_0 \bar{r}_0 - \mathbf{r}_{1:3} \cdot \bar{\mathbf{r}}_{1:3} \\ r_0 \bar{\mathbf{r}}_{1:3} + \bar{r}_0 \mathbf{r}_{1:3} - \mathbf{r}_{1:3} \times \bar{\mathbf{r}}_{1:3} \end{bmatrix} \tag{3.8}$$

definiert. Die Operationen $\mathbf{r}_{1:3} \cdot \bar{\mathbf{r}}_{1:3}$ und $\mathbf{r}_{1:3} \times \bar{\mathbf{r}}_{1:3}$ bezeichnen dabei das Skalarprodukt bzw. das Kreuzprodukt im euklidischen Raum. Eine Rotationsmatrix

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{23} & r_{33} \end{bmatrix} \tag{3.9}$$

und ein Quaternion $\mathbf{r}$ werden über die Beziehungen

$$\mathbf{r} = \frac{1}{2}\begin{bmatrix} \sqrt{r_{11}+r_{22}+r_{33}+1} \\ \operatorname{sign}(r_{32}-r_{23})\sqrt{r_{11}-r_{22}-r_{33}+1} \\ \operatorname{sign}(r_{13}-r_{31})\sqrt{r_{22}-r_{33}-r_{11}+1} \\ \operatorname{sign}(r_{21}-r_{12})\sqrt{r_{33}-r_{11}-r_{22}+1} \end{bmatrix} \tag{3.10a}$$

$$\mathbf{R} = \begin{bmatrix} 2\left(r_0^2+r_1^2\right)-1 & 2\left(r_1r_2-r_0r_3\right) & 2\left(r_1r_2+r_0r_2\right) \\ 2\left(r_1r_2+r_0r_3\right) & 2\left(r_0^2+r_2^2\right)-1 & 2\left(r_1r_2+r_0r_2\right) \\ 2\left(r_1r_3-r_0r_2\right) & 2\left(r_2r_3-r_0r_1\right) & 2\left(r_0^2+r_3^2\right)-1 \end{bmatrix} \tag{3.10b}$$

ineinander umgerechnet [18].

Die Orientierung des Objektkoordinatensystems $(0_o x_o y_o z_o)$ in Bezug auf das Kamerakoordinatensystem $(0_c x_c y_c z_c)$ lässt sich über das Quaternion

$$\mathbf{r}_c^o = \begin{bmatrix} \cos\left(\frac{\psi}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{\psi}{2}\right) \end{bmatrix} \otimes \begin{bmatrix} \cos\left(\frac{\phi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right) \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ 0 \\ \sin\left(\frac{\theta}{2}\right) \\ 0 \end{bmatrix} \tag{3.11}$$

angeben. Die zeitliche Änderungsrate des Quaternions ergibt sich mit dem Vektor der Winkelgeschwindigkeiten des Kamerakoordinatensystems $(0_c x_c y_c z_c)$ $\boldsymbol{\omega}_c = [\omega_{c,x}, \omega_{c,y}, \omega_{c,z}]^{\mathrm{T}}$ zu

$$\dot{\mathbf{r}}_c^o = \frac{1}{2}\mathbf{r}_c^o \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_c \end{bmatrix}, \tag{3.12}$$

welche mit der schiefsymmetrischen Matrix

$$\boldsymbol{\Omega}(\boldsymbol{\omega}_c) = \begin{bmatrix} 0 & -\omega_{c,x} & -\omega_{c,y} & -\omega_{c,z} \\ \omega_{c,x} & 0 & \omega_{c,z} & -\omega_{c,y} \\ \omega_{c,y} & -\omega_{c,z} & 0 & \omega_{c,x} \\ \omega_{c,z} & \omega_{c,y} & -\omega_{c,x} & 0 \end{bmatrix} \tag{3.13}$$

auch über

$$\dot{\mathbf{r}}_c^o = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_c)\mathbf{r}_c^o \tag{3.14}$$

angegeben werden kann. Die Winkelgeschwindigkeiten $\omega_{c,x}, \omega_{c,y}$ und $\omega_{c,z}$ sind die momentanen ortsfesten Drehwinkelgeschwindigkeiten um die x-, y- und z-Achse.

Um die Pose des Objektkoordinatensystems $(0_o x_o y_o z_o)$ in Bezug auf das Kamerakoordinatensystem $(0_c x_c y_c z_c)$ zu vervollständigen, wird der Distanzvektor $\mathbf{d}_c^o$ und dessen zeitliche Ableitung $\dot{\mathbf{d}}_c^o$ definiert. Die Distanz zum Objekt im Basiskoordinatensystem ergibt aus der homogenen Transformation $\mathbf{H}_0^c$ (2.4a) und dem konstanten, jedoch undefinierten Positionsvektor $\mathbf{d}_0^o$ des Objekts im Bezug zum Basiskoordinatensystem zu $\mathbf{d}_c^o = -\mathbf{d}_0^c + \mathbf{d}_0^o$ mit der zeitlichen Ableitung $\dot{\mathbf{d}}_c^o = -\dot{\mathbf{d}}_c^o = -\mathbf{v}_c$.

Die Geschwindigkeiten $\boldsymbol{\omega}_c$ und $\mathbf{v}_c$ werden aus transformierten Sensorgrößen $\tilde{\boldsymbol{\omega}}_c = \left(\mathbf{R}_0^c\right)^{\mathrm{T}}\tilde{\boldsymbol{\omega}}_0^c$ und $\tilde{\mathbf{v}}_c = \tilde{\mathbf{v}}_0^c$ abzüglich einer stochastischen Störung $\mathbf{w}_\omega$ und $\mathbf{w}_v$ über

$$\begin{bmatrix} \boldsymbol{\omega}_c \\ \mathbf{v}_c \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{\omega}}_c - \mathbf{w}_\omega \\ \tilde{\mathbf{v}}_c - \mathbf{w}_v \end{bmatrix} = \begin{bmatrix} \left(\mathbf{R}_0^c\right)^{\mathrm{T}}\tilde{\boldsymbol{\omega}}_0^c - \mathbf{w}_\omega \\ \tilde{\mathbf{v}}_0^c - \mathbf{w}_v \end{bmatrix} \tag{3.15}$$

gebildet. Die Messwerte der Geschwindigkeiten $\tilde{\boldsymbol{\omega}}_0^c$ und $\tilde{\mathbf{v}}_0^c$ in (3.15) bilden sich über die Beziehung (2.6) und (2.7) aus den über Sensorgrößen bestimmten momentanen Gelenkswinkelgeschwindigkeiten $\dot{\mathbf{q}}$ des Roboters.

Damit ergibt sich das Systemmodell des Extended Kalman-Filters mit dem Zustand $\mathbf{x}^{\mathrm{T}} = \left[(\mathbf{r}_c^o)^{\mathrm{T}}, (\mathbf{d}_c^o)^{\mathrm{T}}\right] \in \mathbb{R}^7$, dem Systemeingang $\mathbf{u}^{\mathrm{T}} = \left[\tilde{\boldsymbol{\omega}}_c^{\mathrm{T}}, \tilde{\mathbf{v}}_c^{\mathrm{T}}\right] \in \mathbb{R}^6$, der stochastischen Prozessstörung $\mathbf{w}^{\mathrm{T}} = \left[\mathbf{w}_\omega^{\mathrm{T}}, \mathbf{w}_v^{\mathrm{T}}\right] \in \mathbb{R}^6$, dem Ausgang $\mathbf{y}^{\mathrm{T}} = \left[(\mathbf{r}_c^o)^{\mathrm{T}}, (\mathbf{d}_c^o)^{\mathrm{T}}\right] \in \mathbb{R}^7$ und dem stochastischen Messrauschen $\mathbf{v} \in \mathbb{R}^7$ zu

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{w}) = \begin{bmatrix} \frac{1}{2}\boldsymbol{\Omega}(\tilde{\boldsymbol{\omega}}_c - \mathbf{w}_\omega)\mathbf{r}_c^o \\ -\tilde{\mathbf{v}}_c + \mathbf{w}_v \end{bmatrix}, \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{3.16a}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x},\mathbf{v}) = \begin{bmatrix} \mathbf{r}_c^o \\ \mathbf{d}_c^o \end{bmatrix} + \mathbf{v}. \tag{3.16b}$$

Das Extended Kalman-Filter wird auf einem Digitalrechner mit der Abtastzeit T_a implementiert. Unter der Annahme einer kleinen Abtastzeit, und dass zwischen zwei Abtastzeitpunkten $kT_a \leq t < (k+1)\,T_a$ die Eingangsgrößen konstant mit $\mathbf{u}(t) = \mathbf{u}(kT_a) = \mathbf{u}_k$ sind, ergibt sich aus (3.16) und dem Euler-Verfahren das nichtlineare Abtastsystem

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k,\mathbf{u}_k,\mathbf{w}_k) = \mathbf{x}_k + \mathbf{f}(\mathbf{x}_k,\mathbf{u}_k,\mathbf{w}_k)\,T_a, \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{3.17a}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k,\mathbf{v}_k)\,. \tag{3.17b}$$

Um die Iterationsgleichungen für das Extended Kalman-Filter herzuleiten, wird zunächst für die rechte Seite der Systemdynamik (3.17a) eine Taylor-Reihenentwicklung um den Punkt $\mathbf{x}_k = \hat{\mathbf{x}}_k^+$, $\mathbf{u}_k = \mathbf{u}_k$ und $\mathbf{w}_k = \mathbf{0}$, sowie für die rechte Seite der Ausgangsgleichung (3.17b) eine Taylor-Reihenentwicklung um den Punkt $\mathbf{x}_k = \hat{\mathbf{x}}_k^-$ und $\mathbf{v}_k = \mathbf{0}$ durchgeführt, und jeweils nach dem linearen Term abgebrochen. Daraus resultiert das linearisierte Abtastsystem

$$\mathbf{x}_{k+1} = \boldsymbol{\Phi}_k\mathbf{x}_k + \bar{\mathbf{u}}_k + \mathbf{G}_k\mathbf{w}_k \tag{3.18a}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \breve{\mathbf{u}}_k + \mathbf{v}_k. \tag{3.18b}$$

mit den Systemmatrizen $\boldsymbol{\Phi}_k \in \mathbb{R}^{7\times7}$, $\mathbf{G}_k \in \mathbb{R}^{7\times6}$ und $\mathbf{C}_k \in \mathbb{R}^{7\times7}$, der $n\times n$ Einheitsmatrix

$\mathbf{E}_{n\times n}$ und den Vektoren $\bar{\mathbf{u}}_k \in \mathbb{R}^7$ und $\breve{\mathbf{u}}_k \in \mathbb{R}^7$ gemäß

$$\mathbf{\Phi}_k = \frac{\partial \mathbf{F}\left(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \mathbf{0}\right)}{\partial \mathbf{x}_k} = \begin{bmatrix} \mathbf{E}_{4\times 4} + \frac{1}{2}T_a\mathbf{\Omega}(\mathbf{u}_{k,1:3}) & \mathbf{0}_{4\times 3} \\ \mathbf{0}_{3\times 4} & \mathbf{E}_{3\times 3} \end{bmatrix}, \tag{3.19a}$$

$$\mathbf{G}_k = \frac{\partial \mathbf{F}\left(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \mathbf{0}\right)}{\partial \mathbf{w}_k} = \frac{T_a}{2}\begin{bmatrix} -\hat{\mathbf{x}}_{k,2}^+ & -\hat{\mathbf{x}}_{k,3}^+ & -\hat{\mathbf{x}}_{k,4}^+ & 0 & 0 & 0 \\ \hat{\mathbf{x}}_{k,1}^+ & -\hat{\mathbf{x}}_{k,4}^+ & \hat{\mathbf{x}}_{k,3}^+ & 0 & 0 & 0 \\ \hat{\mathbf{x}}_{k,4}^+ & \hat{\mathbf{x}}_{k,1}^+ & -\hat{\mathbf{x}}_{k,2}^+ & 0 & 0 & 0 \\ -\hat{\mathbf{x}}_{k,3}^+ & \hat{\mathbf{x}}_{k,2}^+ & \hat{\mathbf{x}}_{k,1}^+ & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}, \tag{3.19b}$$

$$\mathbf{C}_k = \frac{\partial \mathbf{h}\left(\hat{\mathbf{x}}_k^-, \mathbf{0}\right)}{\partial \mathbf{x}_k} = \mathbf{E}_{7\times 7}, \tag{3.19c}$$

$$\bar{\mathbf{u}}_k = \mathbf{F}\left(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \mathbf{0}\right) - \mathbf{\Phi}_k\hat{\mathbf{x}}_k^+, \tag{3.19d}$$

$$\breve{\mathbf{u}}_k = \mathbf{h}\left(\hat{\mathbf{x}}_k^-, \mathbf{0}\right) - \mathbf{C}_k\hat{\mathbf{x}}_k^-. \tag{3.19e}$$

Dabei bezeichnet $\mathbf{x}_{k,i}$ bzw. $\mathbf{u}_{k,i}$ das $i$te Element des Zustandes $\mathbf{x}_k$ bzw. des Eingangs $\mathbf{u}_k$. Für die partiellen Ableitungen in (3.19) gilt jeweils die vereinfachte Schreibweise

$$\frac{\partial \mathbf{F}\left(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \mathbf{0}\right)}{\partial \mathbf{x}_k} = \left.\frac{\partial \mathbf{F}\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k\right)}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k=\hat{\mathbf{x}}_k^+, \mathbf{u}_k=\mathbf{u}_k, \mathbf{w}_k=\mathbf{0}} \tag{3.20}$$

Weiters wird für die Störung $\mathbf{w}_k$ und das Messrauschen $\mathbf{v}_k$ vorausgesetzt, dass mit dem Kroneckersymbol $\delta_{kj} = 1$ für $k = j$ und $\delta_{kj} = 0$ sonst gilt

$$\mathrm{E}\left(\mathbf{v}_k\right) = \mathbf{0} \qquad \mathrm{E}\left(\mathbf{v}_k\mathbf{v}_j^{\mathrm{T}}\right) = \mathbf{R}_k\delta_{kj} \tag{3.21a}$$

$$\mathrm{E}\left(\mathbf{w}_k\right) = \mathbf{0} \qquad \mathrm{E}\left(\mathbf{w}_k\mathbf{w}_j^{\mathrm{T}}\right) = \mathbf{Q}_k\delta_{kj} \tag{3.21b}$$

$$\mathrm{E}\left(\mathbf{w}_k\mathbf{v}_j^{\mathrm{T}}\right) = \mathbf{0}. \tag{3.21c}$$

Die Werte von $\mathbf{R}_k$ und $\mathbf{Q}_k$ sind in Tabelle A.5 im Anhang angegeben.

Die gemessene Ausgangsgröße $\mathbf{y}_k$ ist die über den EPnP Algorithmus berechnete Pose $[\mathbf{R}_c^o|\mathbf{t}_c^o]$ aus (3.6), wobei die Rotationsmatrix $\mathbf{R}_c^o$ über (3.10) in das Quaternion $\mathbf{r}_c^o$ umgerechnet, sowie der Vektor $\mathbf{t}_c^o$ im Kamerakoordinatensystem über $\mathbf{d}_c^o = \mathbf{R}_0^c\mathbf{t}_c^o$ ins Basiskoordinatensystem transformiert wird.

Das Kalman-Filter ist über die Iterationsgleichungen

$$\hat{\mathbf{L}}_k = \mathbf{P}_k^- \mathbf{C}_k^\mathrm{T} \left(\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^\mathrm{T} + \mathbf{R}_k\right)^{-1} \tag{3.22a}$$
$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \hat{\mathbf{L}}_k \left(\mathbf{y}_k - \mathbf{C}_k \hat{\mathbf{x}}_k^-\right) \tag{3.22b}$$
$$\mathbf{P}_k^+ = \left(\mathbf{E} - \hat{\mathbf{L}}_k \mathbf{C}_k\right) \mathbf{P}_k^- \tag{3.22c}$$
$$\mathbf{\Phi}_k = \frac{\partial \mathbf{F}\left(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \mathbf{0}\right)}{\partial \mathbf{x}_k} \tag{3.22d}$$
$$\mathbf{G}_k = \frac{\partial \mathbf{F}\left(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \mathbf{0}\right)}{\partial \mathbf{w}_k} \tag{3.22e}$$
$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{F}\left(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \mathbf{0}\right) \tag{3.22f}$$
$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k^+ \mathbf{\Phi}_k^\mathrm{T} + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^\mathrm{T} \tag{3.22g}$$

gegeben mit der a priori Schätzung des Zustandes $\hat{\mathbf{x}}_k^-$ und der Fehlerkovarianzmatrix $\mathbf{P}_k^-$, der a posteriori Schätzung des Zustandes $\hat{\mathbf{x}}_k^+$ und der Fehlerkovarianzmatrix $\mathbf{P}_k^+$ und der Kalman-Verstärkungsmatrix $\hat{\mathbf{L}}_k$. Zu Beginn der Messung wird das Kalman-Filter mit dem gemessenen Zustand $\hat{\mathbf{x}}_0 = \mathbf{y}_k$ und der Fehlerkovarianzmatrix $\mathbf{P}_0 = \mathbf{E}_{7\times7}$ initialisiert. Da zwischen den einzelnen Messwerten etliche ms vergehen und die Abtastzeit T_a wesentlich kürzer ist, wird das Update der Zustandsschätzung (3.22b) sowie das Update der Fehlerkovarianzmatrix (3.22c) nicht in jedem Iterationsschritt durchgeführt, sondern nur wenn ein neuer Messwert $\mathbf{y}_k$ aus (3.6) zur Verfügung steht. Für die Iterationen zwischen zwei Messwerten $\mathbf{y}_k$ gilt $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^-$ sowie $\mathbf{P}_k^+ = \mathbf{P}_k^-$.

Das Kalman-Filter liefert die geschätzte Translation $\hat{\mathbf{t}}_c^o = \left(\mathbf{R}_0^c\right)^\mathrm{T} \hat{\mathbf{d}}_c^o$ und mit (3.10) die geschätzte Rotation $\hat{\mathbf{R}}_c^o$. Die geschätzte Pose des Objekts im Bezugskoordinatensystem ergibt sich schließlich zu

$$\hat{\mathbf{H}}_0^o = \begin{bmatrix} \hat{\mathbf{R}}_0^o & \hat{\mathbf{d}}_0^o \\ \mathbf{0}^\mathrm{T} & 1 \end{bmatrix} \qquad \hat{\mathbf{R}}_0^o = \mathbf{R}_0^c \hat{\mathbf{R}}_c^o \qquad \hat{\mathbf{d}}_0^o = \mathbf{d}_0^c + \mathbf{R}_0^c \hat{\mathbf{t}}_c^o. \tag{3.23}$$

Mit der geschätzten Pose (3.23) des Objekts wird im nächsten Kapitel eine Trajektorienplanung durchgeführt.

4 Trajektorienplanung

Dieses Kapitel beschäftigt sich mit der Berechnung der Trajektorien $\mathbf{q}$, $\dot{\mathbf{q}}$ und $\ddot{\mathbf{q}}$ des Roboters sowie $\mathbf{q}_h$, $\dot{\mathbf{q}}_h$ und $\ddot{\mathbf{q}}_h$ der Greifhand auf Basis von [10, 11]. Die Trajektorien des Roboters und der Greifhand werden über eine Offline-Trajektorienplanung berechnet. Über eine Online-Trajektorienplanung wird während der Roboterbewegung ein Update der Robotertrajektorie durchgeführt. Die Offline- bzw. Online-Planung erfolgt optimalitätsbasiert auf Basis eines zeitkontinuierlichen Integratorsystems, aus welchem durch Diskretisierung ein statisches beschränktes Optimierungsproblem formuliert wird. Durch Interpolation der Lösung des beschränkten statischen Optimierungsproblems wird eine zeitkontinuierliche Trajektorie berechnet.

Vor der Offline-Trajektorienplanung wird in Abschnitt 4.1 der optimale Griff berechnet. Aufbauend auf dem optimalen Griff erfolgt eine Offline-Trajektorienplanung der Finger in Abschnitt 4.2 und anschließend eine Offline-Trajektorienplanung des Roboters in Abschnitt 4.3. Die gesamte Robotertrajektorie setzt sich aus drei einzelnen Trajektorien zusammen, eine Trajektorie für die Bewegung während des Schließens der Finger in Abschnitt 4.3.1, eine für die Bewegung von der Initialposition zum Objekt in Abschnitt 4.3.2 und eine für die Bewegung zur Zielposition nach dem Greifen des Objekts in Abschnitt 4.3.3. In Abschnitt 4.4 werden die berechneten Trajektorien interpoliert. Während der Roboterbewegung wird zyklisch die Pose des Objekts über das Kalman-Filter geschätzt. Anhand dieser Schätzung wird in Abschnitt 4.5 ein online Update der Robotertrajektorie aus Abschnitt 4.3 durchgeführt. Nach jedem online Update erfolgt erneute eine Interpolation der Trajektorie gemäß Abschnitt 4.4.

Die zeitliche Abfolge eines Greifvorgangs ist in Abbildung 4.1 dargestellt. Im blauen Bereich mit $t \in [0,T_1]$ bewegt sich der Roboter von einer Initialposition $\mathbf{q}_0$ zum Objekt. Das Greifen des Objekts findet im roten Bereich mit $t \in [T_1,T_1 + T_2]$ statt. In diesem Bereich wird zusätzlich zur Fingerbewegung mit dem Roboter eine Schwenkbewegung der Greifhand durchgeführt. Im grünen Bereich mit $t \in [T_1 + T_2,T]$ und $T = T_1 + T_2 + T_3$ ist der Greifvorgang beendet und der Roboter bewegt sich zu einer Zielposition $\mathbf{q}_T$.

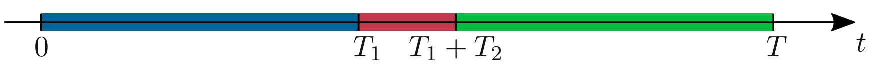


Abbildung 4.1: Zeitlinie des gesamten Greifvorganges.

Abbildung 4.2 zeigt eine Beispielkonfiguration für die Planung einer Trajektorie. Es sind die Koordinatensysteme der Hand, der Kamera, des Objekts, des Greifpunktes am Objekt und das Basiskoordinatensystem eingezeichnet. Die nicht beschrifteten Koordinatensysteme sind die Koordinatensysteme in den Gelenken des Roboters. Zusätzlich sind die homogenen Transformationen $\mathbf{H}_0^c$ zwischen dem Basiskoordinatensystem und der Kamera, $\mathbf{H}_0^h$ zwischen

dem Basiskoordinatensystem und der Greifhand, $\mathbf{H}_o^g$ zwischen dem Koordinatensystem des Objekts und des Greifpunktes, $\hat{\mathbf{H}}_c^o$ zwischen dem Koordinatensystem der Kamera und des Objekts sowie $\hat{\mathbf{H}}_0^o$ zwischen dem Basiskoordinatensystem und des Objekts angegeben. Bei den mit ˆ gekennzeichneten homogenen Transformationen $\hat{\mathbf{H}}_c^o$ und $\hat{\mathbf{H}}_0^o$ handelt es sich um geschätzte Größen auf Basis der Schätzung des Kalman-Filters. Im weiteren Verlauf des Kapitels werden diese als bekannt angenommen und mit $\mathbf{H}_c^o$ und $\mathbf{H}_0^o$ bezeichnet.

Ausgangspunkt der Offline-Trajektorienplanung der Greifhand in Abschnitt 4.2 sowie für den Roboter in Abschnitt 4.3 ist das lineare, zeitinvariante und zeitkontinuirliche Integratorsystem mit den gewünschten Zustandsgrößen $\mathbf{x}_d^\mathrm{T} = \left[\mathbf{q}_d^\mathrm{T}, \dot{\mathbf{q}}_d^\mathrm{T}\right] \in \mathbb{R}^{2n}$ und dem gewünschten Eingang $\mathbf{u}_d = \ddot{\mathbf{q}}_d \in \mathbb{R}^n$ in Systemdarstellung

$$\dot{\mathbf{x}}_d = \mathbf{f}\left(\mathbf{x}_d,\mathbf{u}_d\right) = \begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{E}_{n\times n} \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} \end{bmatrix} \mathbf{x}_d + \begin{bmatrix} \mathbf{0}_{n\times n} \\ \mathbf{E}_{n\times n} \end{bmatrix} \mathbf{u}_d. \tag{4.1}$$

Für den weiteren Verlauf des Kapitels wird für eine bessere Lesbarkeit auf den Index d in den gewünschten Zustandsgrößen und Eingangsgrößen verzichtet. Die Trajektorie vom Anfangszustand $\mathbf{x}_0$ zum Endzustand $\mathbf{x}_N$ mit einer Zeitdauer T wird durch $N+1$ Stützstellen dargestellt. Die Werte an den Stützstellen $k \in \{0,N-1\}$ zu den Zeitpunkten $t = k\Delta t$ und der zeitlichen Schrittweite zwischen den Stützstellen $\Delta t = T_a = T/N$ werden über die sog. Trapez-Methode [19]

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{1}{2}\Delta t\left(\mathbf{f}_k + \mathbf{f}_{k+1}\right) \tag{4.2}$$

mit $\mathbf{f}_k = \mathbf{f}\left(\mathbf{x} = \mathbf{x}_k,\mathbf{u} = \mathbf{u}_k\right)$ aus (4.1) berechnet. Die optimalen Werte der Zustandsgrößen $\mathbf{x}_k^*$ und der Eingangsgrößen $\mathbf{u}_k^*$ an den Stützstellen berechnen sich aus dem beschränkten statischen Optimierungsproblem

$$\min_{\boldsymbol{\xi}} \quad J\left(\boldsymbol{\xi}\right) = T + \Delta t \sum_{k=1}^{N} \mathbf{u}_k^\mathrm{T}\mathbf{u}_k \tag{4.3a}$$

$$\text{u.B.v.} \quad \mathbf{x}_{k+1} - \mathbf{x}_k = \frac{1}{2}\Delta t\left(\mathbf{f}_k + \mathbf{f}_{k+1}\right) \tag{4.3b}$$

$$\mathbf{x}_0 = \mathbf{x}_S, \quad \mathbf{x}_N = \mathbf{x}_T \tag{4.3c}$$

$$\underline{\mathbf{x}} \leq \mathbf{x}_k \leq \overline{\mathbf{x}}, \quad k = 0,\dots,N \tag{4.3d}$$

$$\underline{\mathbf{u}} \leq \mathbf{u}_k \leq \overline{\mathbf{u}}, \quad k = 0,\dots,N \tag{4.3e}$$

mit dem Vektor der Optimierungsvariablen $\boldsymbol{\xi}^\mathrm{T} = \left[T, \mathbf{x}_0^\mathrm{T}, \dots, \mathbf{x}_N^\mathrm{T}, \mathbf{u}_0^\mathrm{T}, \dots, \mathbf{u}_N^\mathrm{T}\right]$ und den Beschränkungen der Zustände und der Eingänge $\underline{\mathbf{x}}$, $\overline{\mathbf{x}}$, $\underline{\mathbf{u}}$ und $\overline{\mathbf{u}}$. In den Abschnitten 4.2 und 4.3 werden insgesamt vier beschränkte statische Optimierungsprobleme der Form (4.3) gelöst. Bei zwei Trajektorien ist jedoch eine gewünschte Trajektoriendauer T_d vorgegeben. Durch das Entfernen der Trajektoriendauer T aus (4.3a) und im Vektor der Optimierungsvariablen $\boldsymbol{\xi}$ kann eine gewünschte Trajektoriendauer $T = T_d$ mit $\Delta t = \frac{T_d}{N}$ vorgegeben werden.

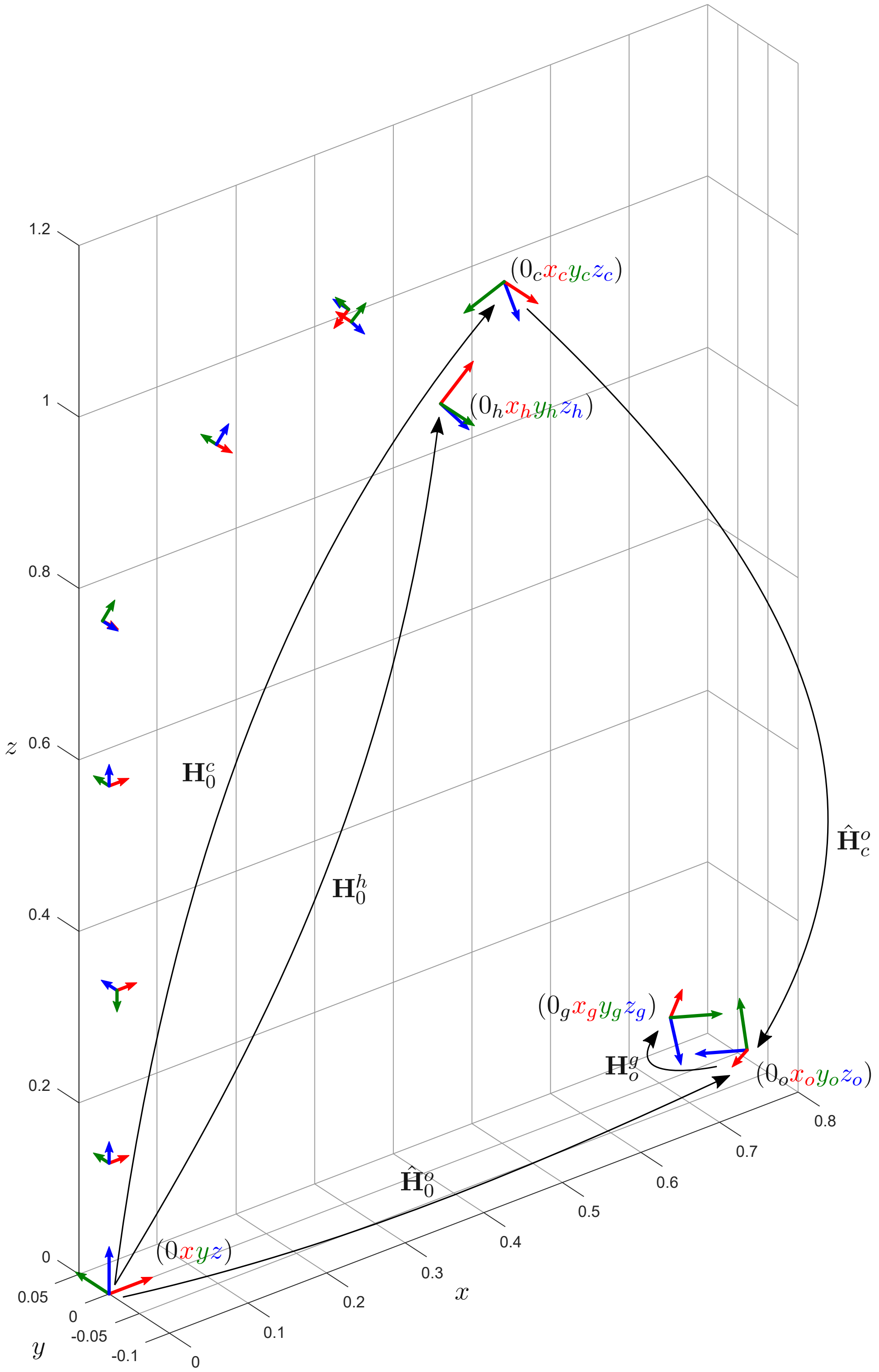


Abbildung 4.2: Beispielanordnung der Koordinatensysteme des Objekts $(0_o x_o y_o z_o)$, des Greifpunktes $(0_g x_g y_g z_g)$, der Hand $(0_h x_h y_h z_h)$ und der Kamera $(0_c x_c y_c z_c)$. Die kleinen unbeschrifteten Koordinatensysteme stellen die Koordinatensysteme $(0_i x_i y_i z_i)$, $i \in \{0, \ldots, 7\}$ des Roboters aus Abbildung 2.1 dar.

4.1 Optimaler Griff

Bevor die einzelnen Trajektorien des Roboters und der Finger berechnet werden können, muss die Pose der Hand zu Beginn und am Ende des Griffs zu den Zeitpunkten $t = T_1$ bzw. $t = T_1 + T_2$ bekannt sein. Hierfür wird in Abschnitt 4.1.1 die Pose des Greifpunkts definiert. Der Greifwinkel wird dabei auf Basis eines Optimierungsproblems berechnet. In Abschnitt 4.1.2 wird zuerst mit dem Greifwinkel die optimale Position und Orientierung des Handkoordinatensystems $(0_h x_h y_h z_h)$ zu Beginn des Griffs berechnet und anschließend die optimalen Zustandsgrößen der Finger $\mathbf{q}_h$ bzw. $\mathbf{q}_G$ am Ende des Griffs. Für die folgenden Berechnungen wird angenommen, dass sich das Greifobjekt zwischen den Fingern befindet, und somit die Greifpunkte g am Objekt und G zwischen den Fingern an der selben Position befinden.

4.1.1 Optimaler Greifwinkel

Um einen Griff zu planen, wird am Objekt aus Abbildung 2.4 ein Greifpunkt, gekennzeichnet durch das Greifkoordinatensystem $(0_g x_g y_g z_g)$, definiert. Die Rotationsmatrix des Koordinatensystems des Greifpunkts in Bezug auf das Koordinatensystem des Objekts $(0_o x_o y_o z_o)$, siehe Abbildung 4.3, ist

$$\mathbf{R}_o^g = \mathbf{R}_o^g(\phi_g) = \mathbf{R}_{z,-\pi}\mathbf{R}_{x,-\frac{\pi}{2}}\mathbf{R}_{y,\phi_g} = \begin{bmatrix} -\cos\phi_g & 0 & -\sin\phi_g \\ \sin\phi_g & 0 & -\cos\phi_g \\ 0 & -1 & 0 \end{bmatrix}, \tag{4.4}$$

wobei der Winkel ϕ_g im nächsten Schritt über ein Optimierungsproblem berechnet wird. Dies ermöglicht es den Abstand zwischen dem Endeffektor des Roboters und der Ebene, auf welche das Objekt platziert wird, zu Beginn des Greifvorgangs zu maximieren.

Die Orientierung der Greifhand $\mathbf{R}_0^h$ zu Beginn des Griffs ist durch die Orientierung des Greifpunktes $\mathbf{R}_0^g$ gegeben, dabei sind die x_h-z_h-Ebene und die x_g-z_g-Ebene deckungsgleich. Des Weiteren wird gefordert, dass die z_h- und z_g-Achse parallel sind und in die gleiche Richtung zeigen. Der Winkel ϕ_g soll nun so festgelegt werden, dass der Abstand zwischen der x-y-Ebene und dem Ursprung des Roboterendeffektors maximal wird. Dadurch soll erreicht werden, dass weder die Kamera, noch die Linse sowie die Greifhand während des Greifvorgangs beschädigt werden. Äquivalent dazu ist das Maximieren der Länge $s_{z,g}$ aus Abbildung 4.3.

Die Länge $s_{z,g}$ berechnet sich mit dem Richtungsvektor $\mathbf{e}_{z,g}$ der z-Achse des Koordinatensystems $(0_g x_g y_g z_g)$ in Bezug zum Basiskoordinatensystem

$$\mathbf{e}_{z,g} = \mathbf{e}_{z,g}(\phi_g) = \mathbf{R}_0^o\mathbf{R}_o^g\mathbf{e}_z. \tag{4.5}$$

zu

$$s_{z,g} = \mathbf{e}_z^{\mathrm{T}}\mathbf{e}_{z,g}. \tag{4.6}$$

Da die Vektoren in (4.6) in entgegengesetzte Richtungen zeigen, ergibt sich für die Länge $s_{z,g}$ ein negativer Wert, was für die weitere Berechnung aber nicht weiter stört. Schließlich

lässt sich das beschränkte Optimierungsproblem

$$\min_{\phi_g} \quad J(\phi_g) = -s_{z,g}^2 \tag{4.7a}$$

$$\text{u.B.v.} \quad s_{z,g} = \mathbf{e}_z^{\mathrm{T}} \mathbf{e}_{z,g} \tag{4.7b}$$

$$\underline{\phi_g} \le \phi_g \le \overline{\phi_g}, \tag{4.7c}$$

für die Berechnung von ϕ_g mit der unteren und oberen Grenze $\underline{\phi_g} = 0°$ und $\overline{\phi_g} = 110°$ formulieren. Die Grenzen für den zulässigen Bereich (4.7c) werden so festgelegt, damit keine Kollisionen zwischen den Fingern und dem Objekt auftreten.

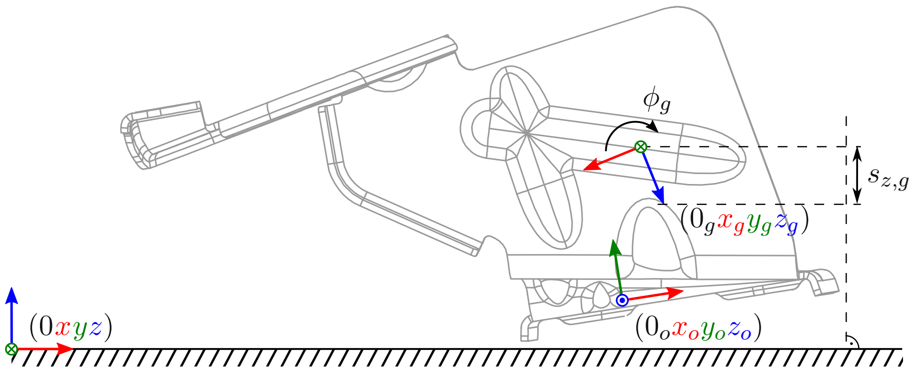


Abbildung 4.3: Seitenansicht des Greifobjekts.

4.1.2 Finaler Greifzustand

Ausgehend von der Lösung zu (4.7) werden nun die neu eingeführten Zustandsgrößen des Griffs $\mathbf{q}_{G,T_2} = [d_{G,T_2}, \varphi_{G,T_2}, y_{G,T_2}, z_{G,T_2}]^{\mathrm{T}}$ zum gewünschten Endzeitpunkt T_2 berechnet. Über die inverse Kinematik (2.37) lassen sich die Zustandsgrößen der Finger $\mathbf{q}_{h,T_2}$ angeben. Das Greifen mit den Fingern wird zum Zeitpunkt $t = T_1$ gestartet und ist zum Zeitpunkt $t = T_1 + T_2$ abgeschlossen. Wie im vorigen Abschnitt beschrieben, sind zu Beginn des Greifvorgangs mit den Fingern zum Zeitpunkt $t = T_1$ die Achsen z_h und z_g parallel und zeigen in die gleiche Richtung. Ein Greifen des Objekts ist ebenfalls möglich, wenn die Greifhand um 180° um die z_h-Achse rotiert wird. Zunächst wird eine Bedingung formuliert, wann eine solche Rotation erforderlich ist. Darauf aufbauend kann im Anschluss der optimale Endzustand $\mathbf{q}_{h,T_2}$ bzw. $\mathbf{q}_{G,T_2}$ der Finger nach dem Griff berechnet werden.

Im ersten Schritt wird ein zusätzliches Koordinatensystem $(0_{\tilde{g}} x_{\tilde{g}} y_{\tilde{g}} z_{\tilde{g}})$ festgelegt, welches um den Winkel π um die z_g-Achse des Koordinatensystems $(0_g x_g y_g z_g)$ gedreht ist, entsprechend

$$\mathbf{R}_o^{\tilde{g}} = \mathbf{R}_o^g \mathbf{R}_{z,\pi}. \tag{4.8}$$

Der Ursprung der beiden Koordinatensysteme $(0_g x_g y_g z_g)$ und $(0_{\tilde{g}} x_{\tilde{g}} y_{\tilde{g}} z_{\tilde{g}})$ befindet sich an der Position

$$\mathbf{d}_0^g = \mathbf{d}_0^{\tilde{g}} = \mathbf{d}_0^o + \mathbf{R}_0^o \mathbf{d}_o^g. \tag{4.9}$$

Zu Beginn des Griffs sind die Greifpunkte g bzw. $\tilde{g}$ und der Greifpunkt G der Hand aus Abbildung 2.7 deckungsgleich. Mit der Annahme, dass die x_h-Achse parallel zu den Achsen x_g bzw. $x_{\tilde{g}}$ ist und jeweils in die gleiche Richtung zeigt, ergeben sich daraus die zwei möglichen Positionen und Rotationen der Hand mit der Vorwärtskinematik (2.25) zu

$$\mathbf{R}_0^h = \mathbf{R}_0^g = \mathbf{R}_0^o \mathbf{R}_o^g, \qquad \mathbf{d}_0^h = \mathbf{d}_0^g - \mathbf{R}_0^h \mathbf{p}_h^G, \tag{4.10a}$$

$$\mathbf{R}_0^{\tilde{h}} = \mathbf{R}_0^{\tilde{g}} = \mathbf{R}_0^o \mathbf{R}_o^{\tilde{g}}, \qquad \mathbf{d}_0^{\tilde{h}} = \mathbf{d}_0^{\tilde{g}} - \mathbf{R}_0^{\tilde{h}} \mathbf{p}_h^G. \tag{4.10b}$$

Ausgehend von (4.10) ergeben sich mit $\mathbf{R}_7^c$, $\mathbf{R}_7^h$, $\mathbf{d}_7^c$ und $\mathbf{d}_7^h$ aus Tabelle A.1, und

$$\mathbf{R}_c^h = \left(\mathbf{R}_7^c\right)^{\mathrm{T}} \mathbf{R}_7^h, \qquad \mathbf{d}_c^h = -\mathbf{d}_7^c + \mathbf{d}_7^h \tag{4.11}$$

die möglichen Positionen und Rotationen des Kamerakoordinatensystems

$$\mathbf{R}_0^c = \mathbf{R}_0^h \left(\mathbf{R}_c^h\right)^{\mathrm{T}}, \qquad \mathbf{d}_0^c = \mathbf{d}_0^h - \mathbf{R}_0^c \mathbf{d}_c^h, \tag{4.12a}$$

$$\mathbf{R}_0^{\tilde{c}} = \mathbf{R}_0^{\tilde{h}} \left(\mathbf{R}_c^h\right)^{\mathrm{T}}, \qquad \mathbf{d}_0^{\tilde{c}} = \mathbf{d}_0^{\tilde{h}} - \mathbf{R}_0^{\tilde{c}} \mathbf{d}_c^h. \tag{4.12b}$$

Die x-Achse des Greifpunktkoordinatensystems wird nun so festgelegt, dass zwischen den beiden Möglichkeiten jene mit dem maximalen Abstand des Kamerakoordinatensystems zum Koordinatensystem der Objektauflagefläche gewählt wird. Damit ergibt sich

$$\mathbf{H}_0^g = \begin{bmatrix} \mathbf{R}_0^i & \mathbf{d}_0^i \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}, \quad \mathbf{R}_0^g = \mathbf{R}_0^i, \quad \mathbf{d}_0^g = \mathbf{d}_0^i, \quad i = \begin{cases} g & \text{wenn } \|\mathbf{d}_0^c\|_2 > \|\mathbf{d}_0^{\tilde{c}}\|_2 \\ \tilde{g} & \text{sonst.} \end{cases} \tag{4.13}$$

Am Ende des Griffs ist das Objekt tangential zur Greiffläche der beiden Finger. Dadurch ist das Koordinatensystem der Hand $(0_h x_h y_h z_h)$ um den Winkel $-\varphi_{G,T_2} = -\varphi_G\,(t = T_2)$ um die x_h-Achse gedreht, entsprechend

$$\mathbf{R}_{0,T_2}^h = \mathbf{R}_0^h\,(t = T_2) = \mathbf{R}_0^h\,(t = T_1)\,\mathbf{R}_{x,-\varphi_{G,T_2}}. \tag{4.14}$$

Der Vektor $\mathbf{p}_{G,T_2}^h$ vom Greifpunkt G zum Koordinatensystem $(0_h x_h y_h z_h)$ im ortsfesten Koordinatensystem $(0_h x_h y_h z_h)$ ergibt sich im Basiskoordinatensystem $(0xyz)$, gekennzeichnet durch den linken Index 0, zum Zeitpunkt $t = T_1 + T_2$ aus

$$\tilde{\mathbf{p}}_{G,T_2}^h = -\mathbf{R}_{0,T_2}^h \mathbf{p}_h^G\,(\mathbf{q}_{G,T_2})\,. \tag{4.15}$$

Ähnlich zur Berechnung des optimalen Greifwinkels ϕ_g in Abschnitt 4.1.1 können aus der z-Komponente des Vektors $\tilde{\mathbf{p}}_{G,T_2}^h$, gegeben durch die Projektion auf die z-Achse

$$s_{z,Gh} = \mathbf{e}_z^{\mathrm{T}} \tilde{\mathbf{p}}_{G,T_2}^h, \tag{4.16}$$

zwei weitere Optimierungsprobleme formuliert werden. Deren Lösung führt jeweils auf optimale Greifzustände, die notwendig sind um das Objekt aufzunehmen. Die Optimierungsprobleme lauten schließlich

$$\min_{\mathbf{q}_{G,T_2}} \quad J\,(\mathbf{q}_{G,T_2}) = \pm s_{z,Gh}^2 \tag{4.17a}$$

$$\text{u.B.v.} \quad -s_{z,Gh} \leq 0 \tag{4.17b}$$

$$\mathbf{h}_G\,(\mathbf{q}_{G,T_2}) \leq \mathbf{0} \tag{4.17c}$$

$$\underline{\mathbf{q}_G} \leq \mathbf{q}_{G,T_2} \leq \overline{\mathbf{q}_G}, \tag{4.17d}$$

$$d_{G,T_2} = 0, \tag{4.17e}$$

mit dem zulässigen Gebiet $\mathbf{h}_G(\mathbf{q}_{G,T_2})$ gemäß (2.46) und den Beschränkungen $\underline{\mathbf{q}_G} = \left[\underline{d}_G, \underline{\varphi}_G, \underline{y}_G, \underline{z}_G\right]^{\mathrm{T}}$ bzw. $\overline{\mathbf{q}_G} = \left[\overline{d}_G, \overline{\varphi}_G, \overline{y}_G, \overline{z}_G\right]^{\mathrm{T}}$. Über das Vorzeichen der Kostenfunktion $J(\mathbf{q}_{G,T_2})$ wird die Richtung der Rotation der Greifhand um die x_g-Achse am Ende des Griffs definiert, womit sich je nach Position und Orientierung des Greifobjekts flüssigere Greifbewegungen realisieren lassen. Die einzelnen Werte der Beschränkungen sind der Tabelle A.8 im Anhang zu entnehmen. Durch die Ungleichungsbeschränkung (4.17b) wird sichergestellt, dass die Distanz (4.16) stets positiv bleibt. Damit ist eine Bedingung für die minimale Steigung der z_h-Achse in Bezug zur x-y-Ebene des Basiskoordinatensystems gegeben. Die Gleichungsbeschränkung (4.17e) gibt an, dass die Finger am Ende vom Griff geschlossen sind. Die optimalen Zustandsgrößen der Finger am Ende vom Griff $\mathbf{q}^*_{h,T_2}$ ergeben sich schließlich aus der inversen Kinematik (2.37). Durch die Ungleichungsbeschränkungen (4.17c) ist sichergestellt, dass die Lösung der inversen Kinematik existiert.

In diesem Abschnitt wurden die optimalen Zustandsgrößen der Finger am Ende des Griffs zum Zeitpunkt $t = T_1 + T_2$ berechnet. Des Weiteren ist über den Initialzustand der Finger $\mathbf{q}_{h,0}$ und der optimalen Lösung $\mathbf{q}^*_{h,T_2}$ mit (4.10a) und (4.13) die optimale Pose der Greifhand $\mathbf{H}^h_0(t = T_1)$ zu Beginn des Greifens zum Zeitpunkt $t = T_1$ sowie die optimale Pose der Greifhand $\mathbf{H}^h_0(t = T_1 + T_2)$ am Ende des Griffs definiert. Darauf aufbauend wird in den nächsten Abschnitten die Trajektorie der Finger sowie des Roboters berechnet.

4.2 Greiftrajektorie der Finger

In diesem Abschnitt wird die Trajektorie für das Greifen des Objekts mit den Fingern berechnet. Als Zeitdauer für das Schließen der Finger wird $t = T_2 = 0{,}5\,\mathrm{s}$ festgelegt. Eine Trajektorie für einen Griff kann in Gelenkskoordinaten $\mathbf{q}_h(t)$ geplant werden, oder aber auch in den Greifkoordinaten $\mathbf{q}_G(t)$ und über die inverse Kinematik in eine Trajektorie der Gelenkskoordinaten umgerechnet werden. Für beide Varianten müssen die finalen optimalen Greifkoordinaten $\mathbf{q}^*_{G,T_2} = \left[d^*_{G,T_2}, \varphi^*_{G,T_2}, y^*_{G,T_2}, z^*_{G,T_2}\right]^{\mathrm{T}}$ bekannt sein. Im Folgenden ist die Planung der Greiftrajektorie direkt in den Gelenkskoordinaten angegeben.

Ausgangspunkt für die Berechnung der Trajektorie in Gelenkskoordinaten ist äquivalent zu (4.1) das lineare, zeitinvariante zeitkontinuierliche Integratorsystem in Systemdarstellung

$$\dot{\mathbf{x}}_h = \mathbf{f}_h(\mathbf{x}_h, \mathbf{u}_h) = \begin{bmatrix} \mathbf{0}_{7\times7} & \mathbf{E}_{7\times7} \\ \mathbf{0}_{7\times7} & \mathbf{0}_{7\times7} \end{bmatrix} \mathbf{x}_h + \begin{bmatrix} \mathbf{0}_{7\times7} \\ \mathbf{E}_{7\times7} \end{bmatrix} \mathbf{u}_h \tag{4.18}$$

mit dem Zustand $\mathbf{x}_h^{\mathrm{T}} = \left[\mathbf{q}_h^{\mathrm{T}}, \dot{\mathbf{q}}_h^{\mathrm{T}}\right] \in \mathbb{R}^{14}$ und dem Eingang $\mathbf{u}_h = \ddot{\mathbf{q}}_h \in \mathbb{R}^7$.

Über eine Darstellung der Trajektorie durch $N_G + 1$ Stützstellen mit der zeitlichen Schrittweite bzw. Abtastzeit $\Delta t_h = \frac{T_2}{N_G}$ und der Trapez-Methode (4.2) ergibt sich für die

Berechnung der Trajektorie ein beschränktes statisches Optimierungsproblem, gemäß

$$\min_{\boldsymbol{\xi}_h} \quad J(\boldsymbol{\xi}_h) = \Delta t_h \sum_{k=1}^{N_G} \mathbf{u}_{h,k}^{\mathrm{T}} \mathbf{u}_{h,k} \tag{4.19a}$$

$$\text{u.B.v.} \quad \mathbf{x}_{h,k+1} - \mathbf{x}_{h,k} = \frac{1}{2} \Delta t_h \left(\mathbf{f}_{h,k} + \mathbf{f}_{h,k+1}\right) \tag{4.19b}$$

$$y_{G,k+1} - y_{G,k} = \frac{y^*_{G,T_2} - y_{G,T_1}}{N_G} \tag{4.19c}$$

$$\mathbf{x}_{h,0} = \mathbf{x}_{h,S}, \quad \mathbf{x}_{h,N_G} = \mathbf{x}_{h,E} \tag{4.19d}$$

$$\underline{\mathbf{x}_h} \leq \mathbf{x}_{h,k} \leq \overline{\mathbf{x}_h}, \quad k = 0, \ldots, N_G \tag{4.19e}$$

$$\underline{\mathbf{u}_h} \leq \mathbf{u}_{h,k} \leq \overline{\mathbf{u}_h}, \quad k = 0, \ldots, N_G \tag{4.19f}$$

mit $\mathbf{f}_{h,k} = \mathbf{f}_h(\mathbf{x}_h = \mathbf{x}_{h,k}, \mathbf{u}_h = \mathbf{u}_{h,k})$, den Vektor der Optimierungsvariablen $\boldsymbol{\xi}_h^{\mathrm{T}} = \left[\mathbf{x}_{h,0}^{\mathrm{T}}, \ldots, \mathbf{x}_{h,N_G}^{\mathrm{T}}, \mathbf{u}_{h,0}^{\mathrm{T}}, \ldots, \mathbf{u}_{h,N_G}^{\mathrm{T}}\right]$, dem Anfangszustand $\mathbf{x}_{h,S}^{\mathrm{T}} = \left[\mathbf{q}_{h,0}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}\right]$, dem Endzustand $\mathbf{x}_{h,E}^{\mathrm{T}} = \left[\left(\mathbf{q}^*_{h,T_2}\right)^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}\right]$ aus Abschnitt 4.1.2, den Beschränkungen der Zustände $\underline{\mathbf{x}_h}$ und $\overline{\mathbf{x}_h}$ und den Beschränkungen der Eingänge $\underline{\mathbf{u}_h}$ und $\overline{\mathbf{u}_h}$. Durch die Gleichungsbeschränkung (4.19c) bewegt sich der Greifpunkt G mit einer gleichmäßigen Geschwindigkeit vom Anfangswert y_{G,T_1} zum optimalen Endwert y^*_{G,T_2}. Durch das Lösen des Optimierungsproblems (4.19) ergeben sich die optimalen Werte der Stützpunkte $\boldsymbol{\xi}_h^*$.

Es wurde die Planung der Greiftrajektorie in den Gelenkskoordinaten $\mathbf{q}_h$ sowie in den Greifkoordinaten $\mathbf{q}_G$ implementiert und getestet. Formal lässt sich das Optimierungsproblem für die Berechnung der Trajektorie in den Greifkoordinaten äquivalent zu (4.19) angeben. Die Implementierung hat aber gezeigt, dass der verwendete Lösungsalgorithmus für das Optimierungsproblem aufgrund der Schrittweite in das nicht zulässige Gebiet springt, wodurch der Algorithmus nicht terminiert und keine Lösung gefunden wurde.

4.3 Roboterbewegung

In diesem Abschnitt wird die Trajektorie des Roboters berechnet. Die gesamte Bewegung des Roboters setzt sich aus drei einzelnen Trajektorien zusammen. Um Sprünge zu vermeiden, wird der Endzustand einer Trajektorie als Anfangszustand der folgenden Trajektorie festgelegt. In Abschnitt 4.3.1 wird die Trajektorie für die Bewegung des Roboters im roten Bereich der Zeitlinie in Abbildung 4.1 während des Greifens des Objekts geplant. Daraus ergeben sich die optimalen Zustandsgrößen des Roboters $\mathbf{q}$ und $\dot{\mathbf{q}}$ zu den Zeitpunkten $t = T_1$ und $t = T_1 + T_2$, mit welchen in Abschnitt 4.3.2 die Robotertrajektorie von der Initialposition zum Objekt im blauen Bereich der Zeitlinie, sowie im Abschnitt 4.3.3 die Trajektorie für die Bewegung zur Zielposition nach dem Greifen des Objekts im grünen Bereich der Zeitlinie geplant wird. Im Folgenden werden die Optimierungsprobleme formuliert, mit denen die drei Trajektorien berechnet werden. Die Formulierungen beruhen auf (4.3), wobei für die Berechnungen der jeweiligen optimalen Trajektorien zusätzliche Ungleichungsbedingungen definiert und berücksichtigt werden.

Ausgangspunkt für die Berechnung der Trajektorien des blauen Zeitbereichs mit $i = 1$, des roten Zeitbereichs mit $i = 2$ und des grünen Zeitbereichs mit $i = 3$ ist äquivalent zu

(4.1) ein lineares, zeitinvariantes zeitkontinuierliches Integratorsystem

$$\dot{\mathbf{x}}_i = \mathbf{f}\left(\mathbf{x}_i,\mathbf{u}_i\right) = \begin{bmatrix} \mathbf{0}_{7\times7} & \mathbf{E}_{7\times7} \\ \mathbf{0}_{7\times7} & \mathbf{0}_{7\times7} \end{bmatrix} \mathbf{x}_i + \begin{bmatrix} \mathbf{0}_{7\times7} \\ \mathbf{E}_{7\times7} \end{bmatrix} \mathbf{u}_i \tag{4.20}$$

mit dem Zustand des Roboters $\mathbf{x}_i^{\mathrm{T}} = \left[\mathbf{q}^{\mathrm{T}}, \dot{\mathbf{q}}^{\mathrm{T}}\right] \in \mathbb{R}^{14}$ und dem Eingang $\mathbf{u}_i = \ddot{\mathbf{q}} \in \mathbb{R}^7$.

4.3.1 Schwenkbewegung während des Greifens

Das Ziel des dynamischen Greifvorgangs des Objektes ist es eine möglichst schnelle, flüssige und zuverlässige Bewegung auszuführen. Die geplante optimale Trajektorie soll also sicherstellen, dass während des Greifvorgangs der Roboter nicht zum Stillstand kommt. Die Geschwindigkeiten $\dot{\mathbf{q}}$ des Roboters am Beginn und am Ende des Griffs sind noch nicht definiert, weshalb zuerst die Robotertrajektorie während des Greifens des Objekts berechnet wird. Beim Schließen der Finger liegen die Greifpunkte G der Finger und g am Objekt übereinander. Es wird angenommen, dass die Normale der Verbindungslinie C_1 und C_3 im Punkt G, siehe Abbildung 2.7, und die z_g-Achse während des Schließens der Finger annähernd parallel sind. Das Objekt liegt somit immer tangential zur Greiffläche der Finger. Daraus ergibt sich, abhängig vom optimalen Endwert φ^*_{G,T_2}, eine Schwenkbewegung der Greifhand, welche über die in diesem Abschnitt berechnete Trajektorie des Roboters durchgeführt wird.

Die Position $\mathbf{d}^G_{0,k} = \mathbf{d}^G_0\left(\mathbf{q}_k\right)$ des Greifpunktes G im Basiskoordinatensystem an den Stützpunkten der Trajektorie ergibt sich aus der Vorwärtskinematik (2.4b) und (2.25) zu

$$\mathbf{d}^G_{0,k} = \mathbf{d}^h_0\left(\mathbf{q}_k\right) + \mathbf{R}^h_0\left(\mathbf{q}_k\right)\mathbf{p}^G_h\left(\mathbf{q}_{h,k}\right). \tag{4.21}$$

Da die Punkte G und g übereinander liegen, sind die Positionen auch über

$$\mathbf{d}^G_{0,k} = \mathbf{d}^g_0 \tag{4.22}$$

gegeben. Der Vektor $\mathbf{d}^g_0$ bildet sich mit der Schätzung des Kalman-Filters aus (4.9). Für die Trajektorie muss (4.22) nicht exakt erfüllt sein, sondern es werden kleine Abweichungen zugelassen. Daraus ergeben sich die Beschränkungen

$$\underline{\Delta\mathbf{d}^G_0} \leq \mathbf{d}^G_{0,k} - \mathbf{d}^g_0 \leq \overline{\Delta\mathbf{d}^G_0} \tag{4.23}$$

mit den unteren und oberen Grenzen $\underline{\Delta\mathbf{d}^G_0}$ und $\overline{\Delta\mathbf{d}^G_0}$. Die experimentell gewählten Grenzen sind der Tabelle A.8 im Anhang zu entnehmen. Dadurch wird der Lösungsraum des Optimierungsproblems erweitert und die Erfolgsrate erhöht, weil der Roboter eine größere Bewegungsfreiheit hat.

Durch die in Abschnitt 4.2 berechnete Trajektorie der Finger ist der Winkel $\varphi^*_{G,k}$ zu jedem Zeitpunkt k bekannt und dadurch auch die Position des Koordinatensystems der Hand

$$\mathbf{R}^h_{0,k} = \mathbf{R}^g_0\mathbf{R}_{x,-\varphi^*_{G,k}}, \tag{4.24a}$$

$$\tilde{\mathbf{d}}^h_{0,k} = \mathbf{d}^g_0 - \mathbf{R}^h_{0,k}\mathbf{p}^G_h\left(\mathbf{q}^*_{h,k}\right) \tag{4.24b}$$

bzw. über die Vorwärtskinematik

$$\tilde{\mathbf{d}}_{0,k}^{h} = \mathbf{d}_0^h(\mathbf{q}_k)\,. \tag{4.25}$$

Äquivalent zu (4.23) werden für (4.25) kleine Abweichungen

$$\underline{\Delta \mathbf{d}_0^h} \leq \tilde{\mathbf{d}}_{0,k}^{h} - \mathbf{d}_0^h(\mathbf{q}_k) \leq \overline{\Delta \mathbf{d}_0^h} \tag{4.26}$$

mit den unteren und oberen Grenzen $\underline{\Delta \mathbf{d}_0^h}$ und $\overline{\Delta \mathbf{d}_0^h}$ zugelassen. Die experimentell gewählten Grenzen für eine größere Bewegungsfreiheit des Roboters sind in Tabelle A.8 im Anhang angegeben.

Die Geschwindigkeit des Koordinatensystems der Hand im Basiskoordinatensystem berechnen sich in jedem Schritt über die Beziehung (2.6) und (2.7) zu

$$\begin{bmatrix} \mathbf{v}_{0,k}^h \\ \boldsymbol{\omega}_{0,k}^h \end{bmatrix} = \mathbf{J}_0^h(\mathbf{q}_k)\,\dot{\mathbf{q}}_k, \tag{4.27}$$

Für die lineare Geschwindigkeit $\mathbf{v}_{0,k}^h$ am Ende vom Griff zum Zeitpunkt $k = N_G$ wird nun gefordert, dass die z-Komponente nicht negativ ist, entsprechend

$$\mathbf{e}_z^{\mathrm{T}} \mathbf{v}_{0,N_G}^h \geq 0, \tag{4.28}$$

oder äquivalent

$$-\mathbf{e}_z^{\mathrm{T}} \mathbf{v}_{0,N_G}^h \leq 0. \tag{4.29}$$

Dadurch wird gewährleistet, dass das Objekt von der Ebene abgehoben wird.

Die Trajektorie bildet sich wie in (4.19) aus $N_G + 1$ Stützstellen mit der Schrittweite $\Delta t = \frac{T_2}{N_G}$. Die optimalen Werte der Stützpunkte der zweiten Trajektorie $\mathbf{x}_{2,k}^*$ und $\mathbf{u}_{2,k}^*$ für $k = 0,1,\ldots,N_G$ errechnen sich schließlich aus dem beschränkten statischen Optimierungsproblem

$$
\begin{aligned}
\min_{\boldsymbol{\xi}_2} \quad & J(\boldsymbol{\xi}_2) = \Delta t \sum_{k=1}^{N} \mathbf{u}_{2,k}^{\mathrm{T}} \mathbf{u}_{2,k} && \text{(4.30a)}\\
\text{u.B.v.} \quad & \mathbf{x}_{2,k+1} - \mathbf{x}_{2,k} = \frac{1}{2}\Delta t\,(\mathbf{f}_k + \mathbf{f}_{k+1}) && \text{(4.30b)}\\
& \mathbf{q}_0 = \mathbf{q}_{S2}, \quad \mathbf{q}_{N_G} = \mathbf{q}_{T2} && \text{(4.30c)}\\
& -\mathbf{e}_z^{\mathrm{T}} \mathbf{v}_{0,N_G}^h \leq 0 && \text{(4.30d)}\\
& \underline{\Delta \mathbf{d}_0^G} \leq \mathbf{d}_{0,k}^G - \mathbf{d}_0^g \leq \overline{\Delta \mathbf{d}_0^G}, \quad k = 0,\ldots,N_G && \text{(4.30e)}\\
& \underline{\Delta \mathbf{d}_0^h} \leq \tilde{\mathbf{d}}_{0,k}^h - \mathbf{d}_0^h(\mathbf{q}_k) \leq \overline{\Delta \mathbf{d}_0^h}, \quad k = 0,\ldots,N_G && \text{(4.30f)}\\
& \underline{\mathbf{x}} \leq \mathbf{x}_{2,k} \leq \overline{\mathbf{x}}, \quad k = 0,\ldots,N_G && \text{(4.30g)}\\
& \underline{\mathbf{u}} \leq \mathbf{u}_{2,k} \leq \overline{\mathbf{u}}, \quad k = 0,\ldots,N_G && \text{(4.30h)}
\end{aligned}
$$

mit der Trapez-Methode (4.30b) und $\mathbf{f}_k = \mathbf{f}(\mathbf{x}_2 = \mathbf{x}_{2,k}, \mathbf{u}_2 = \mathbf{u}_{2,k})$ aus (4.20), dem Vektor der Optimierungsvariablen $\boldsymbol{\xi}_2^{\mathrm{T}} = \left[\mathbf{x}_{2,0}^{\mathrm{T}}, \ldots, \mathbf{x}_{2,N_G}^{\mathrm{T}}, \mathbf{u}_{2,0}^{\mathrm{T}}, \ldots, \mathbf{u}_{2,N_G}^{\mathrm{T}}\right]$, den Beschränkungen

der Zustandsgrößen $\underline{\mathbf{x}}$ und $\overline{\mathbf{x}}$, und den zusätzlichen Beschränkungen (4.23), (4.26) und (4.29). Die Anfangszustände $\mathbf{q}_{S2}$ und Endzustände $\mathbf{q}_{T2}$ werden aus der inversen Kinematik von (2.4b) und (4.24) mit $k \in \{0,N_G\}$ berechnet. Es existieren keine expliziten Werte der Beschränkungen der Eingangsgrößen $\underline{\mathbf{u}}$ und $\overline{\mathbf{u}}$ in (4.30h). Es lassen sich jedoch aus den Bewegungsgleichungen (2.11) mit $\ddot{\mathbf{q}}_k = \mathbf{u}_{2,k}$ und den unteren und oberen Grenzen des Drehmoments $\underline{\boldsymbol{\tau}}$ und $\overline{\boldsymbol{\tau}}$ des Roboters die Bedingungen

$$\underline{\boldsymbol{\tau}} \leq \mathbf{D}\left(\mathbf{q}_k\right)\mathbf{u}_{2,k} + \mathbf{C}\left(\mathbf{q}_k,\dot{\mathbf{q}}_k\right)\dot{\mathbf{q}}_k + \mathbf{g}\left(\mathbf{q}_k\right) \leq \overline{\boldsymbol{\tau}}, \quad k = 0,\dots,N_G \tag{4.31}$$

formulieren und (4.30h) mit dieser ersetzen. Bei der Lösung der Optimierungsaufgabe (4.30) wird durch die aufwendige Berechnung von $\mathbf{C}\left(\mathbf{q}_k,\dot{\mathbf{q}}_k\right)\dot{\mathbf{q}}_k$ in (4.31) die Laufzeit erheblich verlängert. Damit der Einfluss von $\mathbf{C}\left(\mathbf{q}_k,\dot{\mathbf{q}}_k\right)\dot{\mathbf{q}}_k$ auf die Drehmomente abgeschätzt werden kann, wird im Vorfeld eine Monte-Carlo-Analyse mit 10^{10} gleichverteilten Zufallsvariablen $\mathbf{q}_k \in \left[\underline{\mathbf{q}},\overline{\mathbf{q}}\right]$ und $\dot{\mathbf{q}}_k \in \left[\underline{\dot{\mathbf{q}}},\overline{\dot{\mathbf{q}}}\right]$ durchgeführt. Aus der Monte-Carlo-Analyse ergibt sich die Abschätzung

$$\underline{\mathbf{C}} \leq \mathbf{C}\left(\mathbf{q}_k,\dot{\mathbf{q}}_k\right)\dot{\mathbf{q}}_k \leq \overline{\mathbf{C}}. \tag{4.32}$$

Zusammen mit (4.31) ergeben sich schließlich die Ungleichungsbeschränkungen

$$\underline{\boldsymbol{\tau}} - \underline{\mathbf{C}} \leq \mathbf{D}\left(\mathbf{q}_k\right)\mathbf{u}_{2,k} + \mathbf{g}\left(\mathbf{q}_k\right) \leq \overline{\boldsymbol{\tau}} - \overline{\mathbf{C}}, \quad k = 0,\dots,N_G \tag{4.33}$$

welche die Bedingungen (4.30h) ersetzen. Damit wird sichergestellt, dass die Beschränkungen des Roboters eingehalten werden. Die einzelnen Werte der Beschränkungen $\underline{\mathbf{x}}^{\mathrm{T}} = \left[\underline{\mathbf{q}}^{\mathrm{T}},\underline{\dot{\mathbf{q}}}^{\mathrm{T}}\right]$, $\overline{\mathbf{x}}^{\mathrm{T}} = \left[\overline{\mathbf{q}}^{\mathrm{T}},\overline{\dot{\mathbf{q}}}^{\mathrm{T}}\right]$, $\underline{\boldsymbol{\tau}}$, $\overline{\boldsymbol{\tau}}$, $\underline{\mathbf{C}}$ und $\overline{\mathbf{C}}$ sind der Tabelle A.8 im Anhang zu entnehmen.

Die Zustände $\mathbf{x}^*_{2,0}$ und $\mathbf{x}^*_{2,N_G}$ aus der optimalen Lösung $\boldsymbol{\xi}^*_2$ sind jeweils die Anfangs- oder Endzustände für die Trajektorien der folgenden zwei Kapitel.

4.3.2 Von der Initialposition zum Objekt

In diesem Abschnitt wird die Robotertrajektorie im blauen Bereich der Zeitlinie aus Abbildung 4.1 berechnet. Dabei bewegt sich der Roboter vom Initialzustand $\mathbf{x}_0$ zur Anfangsposition des roten Teilabschnitts der Zeitlinie, an dem das Zugreifen mit den Fingern gestartet wird. Die Trajektorie der Roboterzustände wird über ein weiteres statisches beschränktes Optimierungsproblem gemäß (4.3) berechnet. Um möglichst viele Bilder für die Poseschätzung des Objekts während der Roboterbewegung zur Verfügung und unerwünschte Kollisionen der Finger mit dem Objekt zu vermeiden, werden zusätzliche Beschränkungen für das statische beschränkte Optimierungsproblem formuliert.

Während der Greifbewegung ist es notwendig, zyklisch neue Messungen für das Update der Posenschätzung durch das Kalman-Filter zu erhalten. Dafür muss gewährleistet werden, dass sich das Objekt stets im Sichtfeld der Kamera befindet. Dazu wird in jedem Abtastschritt k über (3.3)-(3.5) der Ursprung des Objektkoordinatensystems $(0_o x_o y_o z_o)$ aus der Sicht des Kamerakoordinatensystems $(0_c x_c y_c z_c)$ in den Punkt $\tilde{\mathbf{p}}^o_{c,k}$ in die Bildebene

projiziert, gemäß

$$\begin{bmatrix}\tilde{u}_k\\ \tilde{v}_k\\ \tilde{w}_k\end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x\\ 0 & f_y & c_y\\ 0 & 0 & 1\end{bmatrix}\left[\mathbf{R}^o_{c,k}|\mathbf{d}^o_{c,k}\right] \tag{4.34a}$$

$$\tilde{\mathbf{p}}^o_{c,k} = \frac{1}{\tilde{w}_k}\begin{bmatrix}\tilde{u}_k\\ \tilde{v}_k\end{bmatrix}. \tag{4.34b}$$

Es wird angenommen, dass die geschätzte Pose des Objekts mit $\hat{\mathbf{R}}^o_0$ und $\hat{\mathbf{d}}^o_0$ konstant ist. Die Rotation $\mathbf{R}^o_{c,k}$ und Translation $\mathbf{d}^o_{c,k}$ zwischen dem Kamerakoordinatensystem und dem Objektkoordinatensystem ergibt sich damit in jedem Abtastschritt k zu

$$\mathbf{H}^o_{c,k} = \left(\mathbf{H}^c_0\left(\mathbf{q}_k\right)\right)^{-1}\hat{\mathbf{H}}^o_0 \tag{4.35}$$

mit der homogenen Transformation (2.4). Das Kamerabild hat eine Auflösung von 1024 × 1024 Pixel, woraus sich die Beschränkungen

$$\underline{\tilde{\mathbf{p}}^o_c} \leq \tilde{\mathbf{p}}^o_{c,k} \leq \overline{\tilde{\mathbf{p}}^o_c} \tag{4.36}$$

für das Objektkoordinatensystem in der Bildebene ergeben.

Durch die schnelle Bewegung der Kamera kann es dazu kommen, dass die Bilder aufgrund der Bewegungsunschärfe unbrauchbar für die Schätzung der Pose des Objekts werden. Damit solche unscharfen Bilder durch die schnelle Kameraführung vermieden werden, wird die Änderung des Punktes $\tilde{\mathbf{p}}^o_c$ zwischen den einzelnen Schritten k durch

$$\underline{\Delta\tilde{\mathbf{p}}^o_c} \leq \tilde{\mathbf{p}}^o_{c,k+1} - \tilde{\mathbf{p}}^o_{c,k} \leq \overline{\Delta\tilde{\mathbf{p}}^o_c} \tag{4.37}$$

begrenzt. Die oberen und unteren Grenzen $\underline{\tilde{\mathbf{p}}^o_c}$, $\overline{\tilde{\mathbf{p}}^o_c}$, $\underline{\Delta\tilde{\mathbf{p}}^o_c}$ sowie $\overline{\Delta\tilde{\mathbf{p}}^o_c}$ sind in der Tabelle A.8 des Anhangs zu entnehmen.

Um Kollisionen zu vermeiden und gleichzeitig die Richtung zu definieren, aus der sich der Greifpunkt der Finger G dem Greifpunkt am Objekt g annähert, wird die invertierte und verschobene zweidimensionale Gaußglocke

$$f_G\left(x,y\right) = z_G - \frac{h_{z,0}}{\sqrt{(2\pi)^2\det\boldsymbol{\Sigma}}}\exp\left(-\frac{1}{2}\left(\begin{bmatrix}x\\ y\end{bmatrix} - \boldsymbol{\mu}\right)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left(\begin{bmatrix}x\\ y\end{bmatrix} - \boldsymbol{\mu}\right)\right) \tag{4.38}$$

definiert. Dabei bezeichnet $h_{z,0}$ den Skalierungsfaktor, $\boldsymbol{\Sigma}$ die Kovarianzmatrix und $\boldsymbol{\mu}$ den Vektor der Erwartungswerte der Glockenkurve

$$h_{z,0} = z_G 2\pi\sqrt{\det\left(\boldsymbol{\Sigma}\right)}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix}\sigma_x^2 & 0\\ 0 & \sigma_y^2\end{bmatrix}, \qquad \boldsymbol{\mu} = \begin{bmatrix}0\\ 0\end{bmatrix}. \tag{4.39}$$

Die Höhe der Kurve lässt sich mit z_G einstellen. Die Funktion $f_G\left(x,y\right) \in [0,z_G]$ aus (4.38) mit $z_G = 0{,}2\,\mathrm{m}$ ist in Abbildung 4.4 dargestellt. Durch die Varianzen σ_x^2 und σ_y^2 wird die Breite der Gauß-Glocke in x- und y-Richtung verändert. Die Varianzen wurden zu $\sigma_x^2 = 5\,\mathrm{mm}$ und $\sigma_y^2 = 2\,\mathrm{mm}$ gewählt. Das Koordinatensystem der Gauß-Glocke

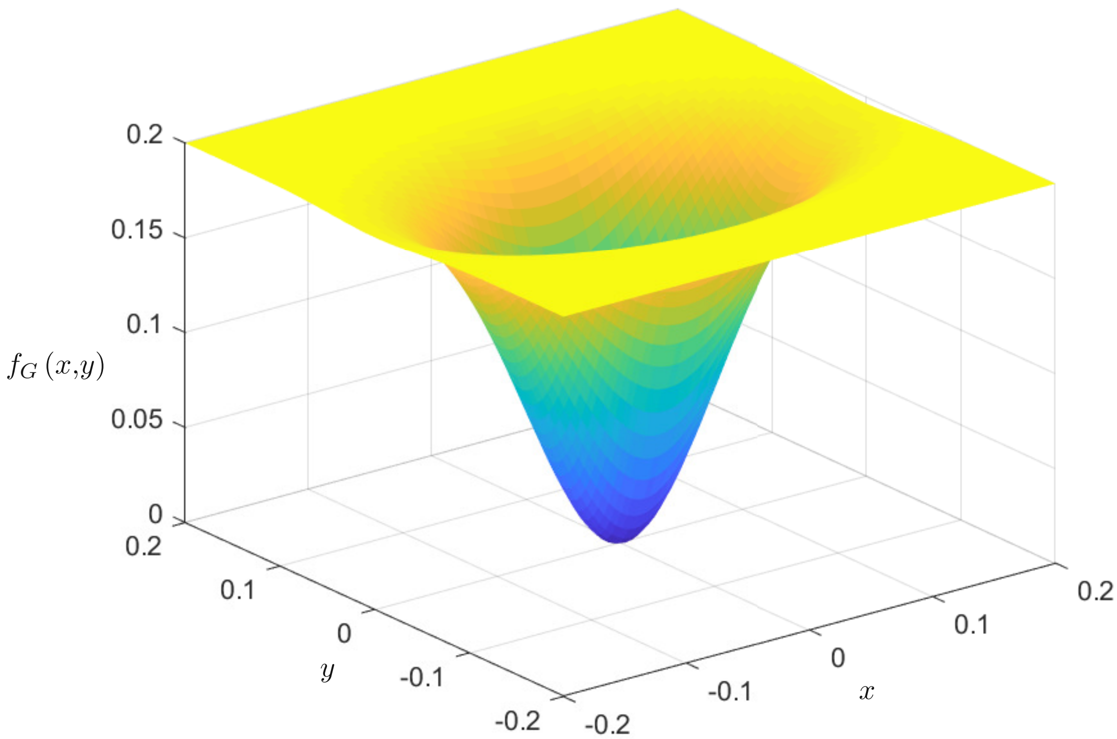


Abbildung 4.4: Zweidimensionale Gauß-Glocke mit $z_G = 0{,}2\,\mathrm{m}$.

$(0_{GG}x_{GG}y_{GG}z_{GG})$ wird durch die Rotation $\mathbf{R}_0^{GG}$ und die Verschiebung $\mathbf{d}_0^{GG}$ mit

$$\mathbf{R}_0^{GG} = \mathbf{R}_0^g \mathbf{R}_{x_g,\pi} \mathbf{R}_{z_g,\pi}, \qquad \mathbf{d}_0^{GG} = \mathbf{d}_0^g \tag{4.40}$$

in den Greifpunkt g des Objektes gelegt. Durch die Rotation $\mathbf{R}_0^{GG}$ sind die z-Achsen der Koordinatensysteme $(0_g x_g y_g z_g)$ des Greifpunktes und $(0_{GG}x_{GG}y_{GG}z_{GG})$ der Gauß-Glocke deckungsgleich, jedoch entgegengesetzt orientiert. Durch die invertierte und in den Ursprung des Greifpunktes verschobene Gauß-Glocke aus (4.38) wird nun eine Fläche definiert. Für die gesamte Trajektorie im blauen Bereich der Zeitlinie wird nun gefordert, dass der Greifpunkt der Finger G stets über der Fläche der Gauß-Glocke bleibt. Dafür wird zunächst im Basiskoordinatensystem mit (4.21) der Vektor vom Ursprung des Koordinatensystems der Gauß-Glocke zum Greifpunkt G gebildet und in das Koordinatensystem der Gauß-Glocke transformiert

$$\mathbf{p}_{GG}^{G}(\mathbf{q},\mathbf{q}_h) = \begin{bmatrix} x_{GG-G}(\mathbf{q},\mathbf{q}_h) \\ y_{GG-G}(\mathbf{q},\mathbf{q}_h) \\ z_{GG-G}(\mathbf{q},\mathbf{q}_h) \end{bmatrix} = \left(\mathbf{R}_0^{GG}\right)^{\mathrm{T}} \left(-\mathbf{d}_0^{GG} + \mathbf{d}_0^h(\mathbf{q}) + \mathbf{R}_0^h(\mathbf{q})\,\mathbf{p}_h^G(\mathbf{q}_h)\right). \tag{4.41}$$

Dort lässt sich nun mit (4.38) und (4.41) die Distanz

$$s_{z,G-GG}(\mathbf{q},\mathbf{q}_h) = z_{GG-G}(\mathbf{q},\mathbf{q}_h) - f_G\left(x_{GG-G}(\mathbf{q},\mathbf{q}_h), y_{GG-G}(\mathbf{q},\mathbf{q}_h)\right) \tag{4.42}$$

definieren, die den Abstand zwischen der Fläche der Gauß-Kurve und dem Greifpunkt der Finger angibt. Die Beschränkung zur Vermeidung von Kollisionen lautet schließlich

$$-\,s_{z,G-GG}(\mathbf{q},\mathbf{q}_h) \leq 0. \tag{4.43}$$

Abbildung 4.5 zeigt ein Beispiel der Entfernung $s_{z,G-GG}$, in dem sich der Greifpunkt der Finger G auf der y_g-z_g-Ebene in der näheren Umgebung des Greifpunktes g befindet. Die Schnittkurve $f_G(0,y_g)$ der Gauß-Glocke mit der y_g-z_g-Ebene ist ebenfalls eingezeichnet.

Die ersten Trajektorie bildet sich aus N Stützstellen mit der Schrittweite $\Delta t = \frac{T_1}{N}$. Die optimalen Werte der Stützpunkte $\mathbf{x}_{1,k}^*$ und $\mathbf{u}_{1,k}^*$ für $k = 0{,}1\ldots N$ der ersten Trajektorie sowie die optimale Trajektoriendauer T_1^* ergeben sich schließlich aus dem beschränkten statischen Optimierungsproblem

$$\min_{\boldsymbol{\xi}_1} \quad J(\boldsymbol{\xi}_1) = T_1 + \Delta t \sum_{k=1}^{N} \mathbf{u}_{1,k}^{\mathrm{T}} \mathbf{u}_{1,k} \tag{4.44a}$$

$$\text{u.B.v.} \quad \mathbf{x}_{1,k+1} - \mathbf{x}_{1,k} = \frac{1}{2}\Delta t\,(\mathbf{f}_k + \mathbf{f}_{k+1}) \tag{4.44b}$$

$$\mathbf{x}_{1,0} = \mathbf{x}_{S1}, \quad \mathbf{x}_{1,N} = \mathbf{x}_{T1} \tag{4.44c}$$

$$-\,s_{z,G-GG}(\mathbf{q}_k,\mathbf{q}_{h,0}) \leq 0, \quad k = 0,\ldots,N \tag{4.44d}$$

$$\underline{\tilde{\mathbf{p}}_c^o} \leq \tilde{\mathbf{p}}_{c,k}^o \leq \overline{\tilde{\mathbf{p}}_c^o}, \quad k = 0,\ldots,N \tag{4.44e}$$

$$\underline{\Delta\tilde{\mathbf{p}}_c^o} \leq \tilde{\mathbf{p}}_{c,k+1}^o - \tilde{\mathbf{p}}_{c,k}^o \leq \overline{\Delta\tilde{\mathbf{p}}_c^o}, \quad k = 0,\ldots,N-1 \tag{4.44f}$$

$$\underline{\mathbf{x}} \leq \mathbf{x}_{1,k} \leq \overline{\mathbf{x}}, \quad k = 0,\ldots,N \tag{4.44g}$$

$$\underline{\boldsymbol{\tau}} - \underline{\mathbf{C}} \leq \mathbf{D}(\mathbf{q}_k)\,\mathbf{u}_{1,k} + \mathbf{g}(\mathbf{q}_k) \leq \overline{\boldsymbol{\tau}} - \overline{\mathbf{C}}, \quad k = 0,\ldots,N \tag{4.44h}$$

$$\underline{T_1} \leq T_1 \leq \overline{T_1} \tag{4.44i}$$

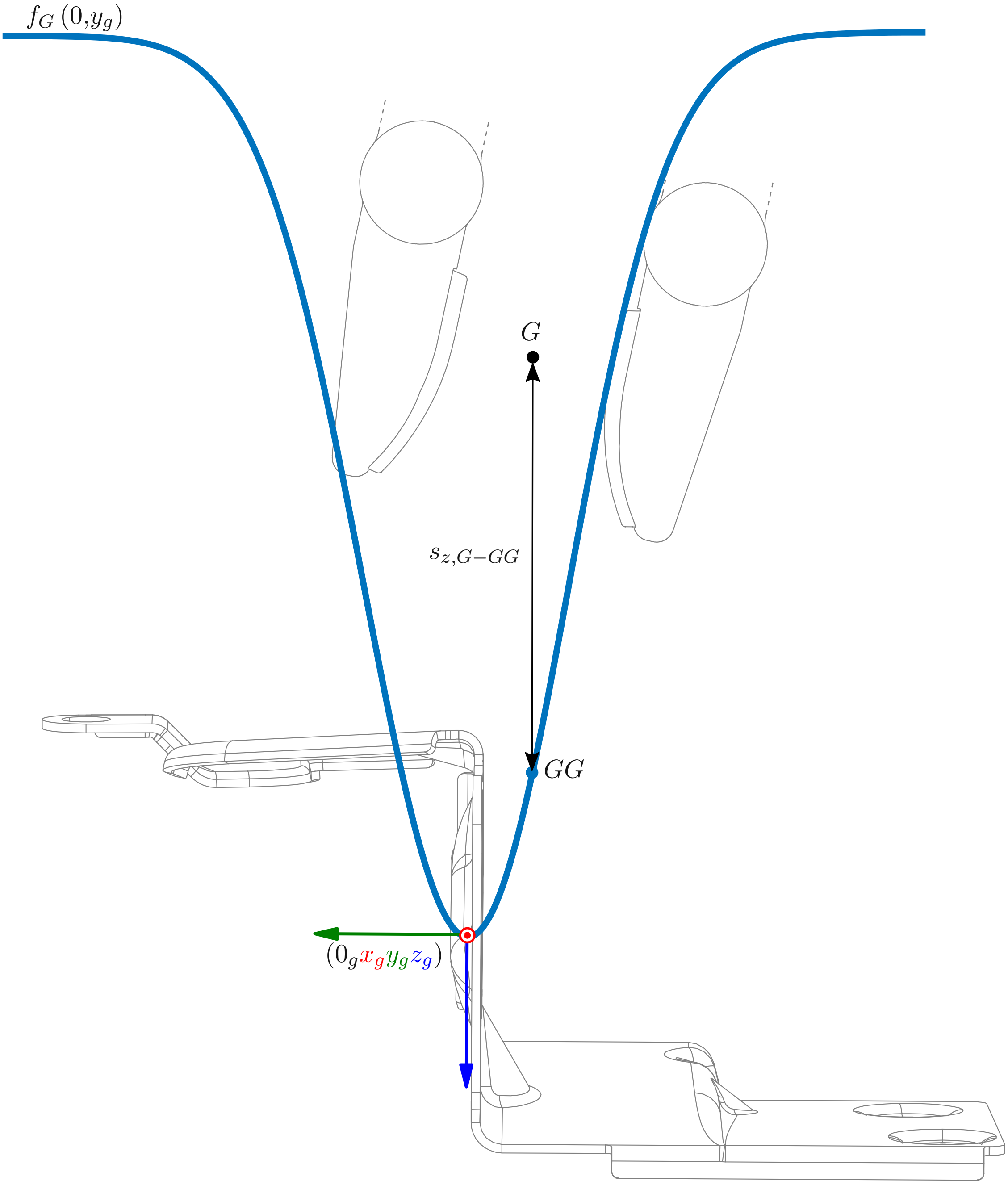


Abbildung 4.5: Seitenansicht des Greifobjekts und der Fingerspitzen mit eingezeichneter Begrenzungskurve $f_G(0,y_o)$ durch die Gauß-Glocke.

mit der Trapez-Methode (4.44b) und $\mathbf{f}_k = \mathbf{f}(\mathbf{x}_1 = \mathbf{x}_{1,k}, \mathbf{u}_1 = \mathbf{u}_{1,k})$ aus (4.20), dem Vektor der Optimierungsvariablen $\boldsymbol{\xi}_1^{\mathrm{T}} = \left[T_1, \mathbf{x}_{1,0}^{\mathrm{T}}, \ldots, \mathbf{x}_{1,N}^{\mathrm{T}}, \mathbf{u}_{1,0}^{\mathrm{T}}, \ldots, \mathbf{u}_{1,N}^{\mathrm{T}}\right]$, dem Anfangszustand $\mathbf{x}_{S1}^{\mathrm{T}} = \left[\mathbf{q}_0^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}\right]$, dem Endzustand $\mathbf{x}_{T1} = \mathbf{x}_{2,0}^*$ aus der Lösung $\boldsymbol{\xi}_2^*$ von (4.30), und den zusätzlichen Beschränkungen (4.36), (4.37) und (4.43). Die Werte aller Beschränkungen sind in Tabelle A.8 angegeben.

4.3.3 Zur Endposition nach dem Greifen

Die dritte und letzte Trajektorie im grünen Bereich der Zeitlinie in Abbildung 4.1 beschreibt die Bewegung des Roboters vom Ende des Griffs zur Zielposition. Für die gesamte Trajektorie wird gefordert, dass in jedem Abtastschritt k die z-Komponente der translatorischen Geschwindigkeit des Endeffektors $\mathbf{v}_{0,k}^h$ aus (4.27) nicht negativ ist, woraus sich die Beschränkung

$$-\mathbf{e}_z^{\mathrm{T}} \mathbf{v}_{0,k}^h \leq 0 \tag{4.45}$$

ergibt. Damit wird erreicht, dass sich das Objekt nicht mehr auf die Ebene zubewegt, auf der es am Anfang gelegen ist.

Die dritte Trajektorie bildet sich äquivalent zu (4.44) aus N Stützstellen mit der Schrittweite $\Delta t = \frac{T_3}{N}$. Die optimalen Werte der Stützpunkte $\mathbf{x}_{3,k}^*$ und $\mathbf{u}_{3,k}^*$ für $k = 0{,}1 \ldots N$ der dritten Trajektorie sowie die optimale Trajektoriendauer T_3^* ergeben sich schließlich aus dem beschränkten statischen Optimierungsproblem

$$\min_{\boldsymbol{\xi}_3} \quad J(\boldsymbol{\xi}_3) = T_3 + \Delta t \sum_{k=1}^{N} \mathbf{u}_{3,k}^{\mathrm{T}} \mathbf{u}_{3,k} \tag{4.46a}$$

$$\text{u.B.v.} \quad \mathbf{x}_{3,k+1} - \mathbf{x}_{3,k} = \frac{1}{2} \Delta t \left(\mathbf{f}_k + \mathbf{f}_{k+1}\right) \tag{4.46b}$$

$$\mathbf{x}_{3,0} = \mathbf{x}_{S3}, \quad \mathbf{x}_{3,N} = \mathbf{x}_{T3} \tag{4.46c}$$

$$-\mathbf{e}_z^{\mathrm{T}} \mathbf{v}_{0,k}^h \leq 0, \quad k = 0, \ldots, N \tag{4.46d}$$

$$\underline{\mathbf{x}} \leq \mathbf{x}_{3,k} \leq \overline{\mathbf{x}}, \quad k = 0, \ldots, N \tag{4.46e}$$

$$\underline{\boldsymbol{\tau}} - \underline{\mathbf{C}} \leq \mathbf{D}(\mathbf{q}_k) \mathbf{u}_{3,k} + \mathbf{g}(\mathbf{q}_k) \leq \overline{\boldsymbol{\tau}} - \overline{\mathbf{C}}, \quad k = 0, \ldots, N \tag{4.46f}$$

$$\underline{T_3} \leq T_3 \leq \overline{T_3} \tag{4.46g}$$

mit der Trapez-Methode (4.46b) und $\mathbf{f}_k = \mathbf{f}(\mathbf{x}_3 = \mathbf{x}_{3,k}, \mathbf{u}_3 = \mathbf{u}_{3,k})$ aus (4.20), dem Vektor der Optimierungsvariablen $\boldsymbol{\xi}_3^{\mathrm{T}} = \left[T_3, \mathbf{x}_{3,0}^{\mathrm{T}}, \ldots, \mathbf{x}_{3,N}^{\mathrm{T}}, \mathbf{u}_{3,0}^{\mathrm{T}}, \ldots, \mathbf{u}_{3,N}^{\mathrm{T}}\right]$, dem Anfangszustand $\mathbf{x}_{S3} = \mathbf{x}_{2,N_G}^*$ aus der Lösung $\boldsymbol{\xi}_2^*$ von (4.30), dem Endzustand $\mathbf{x}_{T3}^{\mathrm{T}} = \left[\mathbf{q}_T^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}\right]$, und der zusätzlichen Beschränkung (4.45). Die Werte aller Beschränkungen sind in Tabelle A.8 angegeben.

4.4 Interpolation der Stützpunkte

In diesem Abschnitt wird eine Interpolation der Stützpunkte der Trajektorien aus Abschnitt 4.2 und 4.3 durchgeführt. Eine Interpolation ist notwendig, da für die Regelung eine wesentlich feinere Auflösung benötigt wird. Von einer Vergrößerung der Anzahl der

Optimierungsvariablen ist abzuraten, da damit die Rechenzeit stark ansteigen würde. Für jeden einzelnen Zeitbereich aus Abbildung 4.1 werden die zugehörigen optimalen Stützpunkte der Trajektorie $\boldsymbol{\xi}_i^*, i \in \{1{,}2{,}3\}$ sowie $\boldsymbol{\xi}_h^*$ über das jeweilige zugehörige Zeitintervall $t \in [0{,}T_1]$, $t \in [T_1{,}T_1 + T_2]$ bzw. $t \in [T_1 + T_2{,}T]$ mit so genannten B-Spline Kurven interpoliert. Somit wird für jede der vier Trajektorien der einzelnen Zeitbereiche getrennt eine Interpolation durchgeführt. Die Implementierung der B-Spline-Interpolation wird anhand von [20] durchgeführt.

Eine B-Spline-Kurve mit vorgegebenen Ableitungen an den Rändern $\mathbf{C}_{NU}\left(\bar{t}\right), \bar{t} \in [0{,}1]$ der Ordnung p, die jeweils durch die N bzw. N_G Stützstellen in $\boldsymbol{\xi}_i^*, i \in \{1{,}2{,}3\}$ bzw. $\boldsymbol{\xi}_h^*$ der Kapitel 4.2 und 4.3 läuft, bildet sich über

$$\mathbf{C}_{NU}\left(\bar{t}\right) = \sum_{i=0}^{N+2} B_{i,p}\left(\bar{t}\right) \mathbf{P}_i \tag{4.47}$$

mit den sog. Basisfunktionen $B_{i,p}$ der Ordnung p und dem Kontrollpunktvektor $\mathbf{P}_i, i \in \{0, \ldots, N+2\}$. Die Ordnung p wird je nach gewünschter stetiger Differenzierbarkeit der Kurve (4.47) gewählt. Für eine r-fach stetig differenzierbare Kurve gilt für die Ordnung

$$p \geq r + 1. \tag{4.48}$$

In dieser Arbeit soll die Kurve zweimal stetig differenzierbar sein, also $r = 2$ und somit ergibt sich die Ordnung

$$p = r + 1 = 3. \tag{4.49}$$

Für die Berechnung der Basisfunktionen wird der Vektor der normierten Zeit der Trajektorie $\bar{\mathbf{t}} = [\bar{t}_0, \ldots, \bar{t}_N]$ mit

$$\bar{t}_0 = 0, \tag{4.50a}$$

$$\bar{t}_N = 1, \tag{4.50b}$$

$$\bar{t}_i = \frac{i}{N}, \quad i = 1, \ldots, N-1 \tag{4.50c}$$

gebildet. Zusätzlich wird der sog. Knotenvektor $\mathbf{t}_K = \left[t_{K,0}, \ldots t_{K,(N+2)+p+1}\right]$ definiert. Die Elemente $t_{K,i}$, $i \in \{0, \ldots, N+p+3\}$ von $\mathbf{t}_K$ werden als Knoten bezeichnet und es gilt

$$t_{K,i} \leq t_{K,i+1}, \quad i = 0, \ldots, N+p+2. \tag{4.51}$$

Die Elemente des Knotenvektors $\mathbf{t}_K$ sind

$$t_{K,0} = \cdots = t_{K,p} = 0 \tag{4.52a}$$

$$t_{K,N+3} = \cdots = t_{K,N+p+3} = 1 \tag{4.52b}$$

$$t_{K,j+p+1} = \frac{1}{p} \sum_{i=j}^{j+p-1} \bar{t}_i, \quad j = 0, \ldots, N-p+1 \tag{4.52c}$$

Mit den Elementen des Knotenvektors $t_{K,i}, i \in \{0,\ldots,N+p+3\}$ sind die N Basisfunktionen $B_{i,p}$ durch

$$B_{i,0}\left(\bar{t}\right) = \begin{cases} 1, & \text{wenn } t_{K,i} \leq \bar{t} < t_{K,i+1} \\ 0, & \text{sonst} \end{cases} \tag{4.53a}$$

$$B_{i,p}\left(\bar{t}\right) = \frac{\bar{t} - t_{K,i}}{t_{K,i+p} - t_{K,i}} B_{i,p-1}\left(\bar{t}\right) + \frac{t_{K,i+p+1} - \bar{t}}{t_{K,i+p+1} - t_{K,i+1}} B_{i+1,p-1}\left(\bar{t}\right) \tag{4.53b}$$

definiert. Die Basisfunktionen $B_{i,j}\left(\bar{t}\right), j \in \{0,1,2,3\}$ sind in Abbildung 4.6 dargestellt. Die Ordnung p bestimmt den Einfluss eines einzelnen Kontrollpunktes auf die Teilbereiche der Kurve. Für die Bereiche $\bar{t} < t_{K,i}$ und $\bar{t} >= t_{K,i+p+1}$ gilt $B_{i,p}(\bar{t}) = 0$.

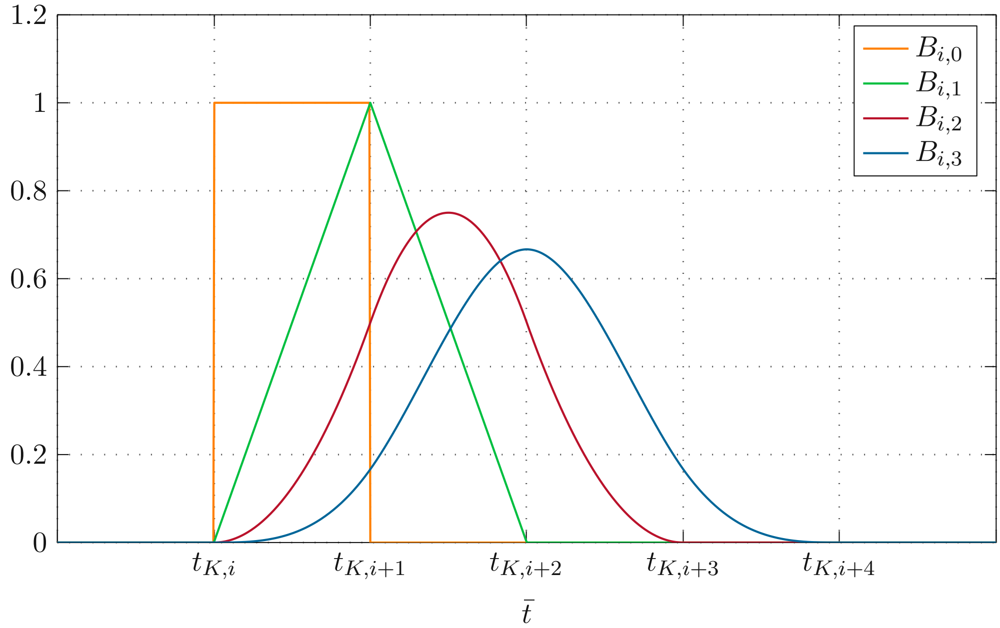


Abbildung 4.6: B-Spline Basisfunktionen $B_{i,j}\left(\bar{t}\right), j \in \{0,1,2,3\}$

Die Ableitungen der B-Spline-Kurve an den Endpunkten sind in der Form

$$\left.\frac{\mathrm{d}}{\mathrm{d}\bar{t}}\mathbf{C}_{NU}\left(\bar{t}\right)\right|_{\bar{t}=0} = \frac{p}{t_{K,p+1}}\left(\mathbf{P}_1 - \mathbf{P}_0\right), \tag{4.54a}$$

$$\left.\frac{\mathrm{d}}{\mathrm{d}\bar{t}}\mathbf{C}_{NU}\left(\bar{t}\right)\right|_{\bar{t}=1} = \frac{p}{1 - t_{K,N+2}}\left(\mathbf{P}_{N+2} - \mathbf{P}_{N+1}\right) \tag{4.54b}$$

gegeben.

Um die einzelnen Trajektorien durch eine B-Spline-Kurve nach (4.47) zu interpolieren, wird zunächst die Zeitachse der jeweiligen Trajektorie t auf die jeweilige Trajektoriendauer $T = T_i, i \in \{1,2,3\}$ normiert, gemäß

$$t' = \frac{t}{T}. \tag{4.55}$$

Das Argument der Kurve (4.47) wird nun durch die normierte Zeit t' ersetzt. Aus den N einzelnen Stützstellen der vorigen Kapitel bilden sich die Punktvektoren $\mathbf{Q}_k^{\mathrm{T}} = \left[\mathbf{q}_k^{\mathrm{T}}, \dot{\mathbf{q}}_k^{\mathrm{T}}, \mathbf{u}_k^{\mathrm{T}}\right], k \in 0, \ldots, N$. Die Punkte $\mathbf{Q}_k$ berechnen sich aus der B-Spline-Kurve aus

$$\mathbf{Q}_k = \mathbf{C}_{NU}\left(t'_k\right) = \sum_{i=0}^{N+2} B_{i,p}\left(t'_k\right)\mathbf{P}_i, \tag{4.56}$$

wobei durch die Normierung der Zeit für die Zeitpunkte $t'_k = \bar{t}_k$, $k \in \{0, \ldots, N\}$ gilt. Für die zeitliche Ableitungen $\dot{\mathbf{Q}}_k^{\mathrm{T}} = [\dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k, \dot{\mathbf{u}}_k]$ an den stellen $k \in \{0, N\}$ ergibt sich mit (4.54)

$$-\mathbf{P}_0 + \mathbf{P}_1 = \frac{t_{K,p+1}}{p}\dot{\mathbf{Q}}_0, \tag{4.57a}$$

$$-\mathbf{P}_{N+1} + \mathbf{P}_{N+2} = \frac{1 - t_{K,N+2}}{p}\dot{\mathbf{Q}}_N. \tag{4.57b}$$

Aus den N Gleichungen (4.56) und den zwei Gleichungen (4.57) lässt sich das lineare Gleichungssystem

$$\mathbf{B}_K\mathbf{P}_K = \mathbf{Q}_K \tag{4.58}$$

mit

$$\mathbf{P}_K = \begin{bmatrix} \mathbf{P}_0^{\mathrm{T}} \\ \mathbf{P}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{P}_{N+2}^{\mathrm{T}} \end{bmatrix}, \qquad \mathbf{Q}_K = \begin{bmatrix} \mathbf{Q}_0^{\mathrm{T}} \\ \frac{t_{K,p+1}}{p}\dot{\mathbf{Q}}_0^{\mathrm{T}} \\ \mathbf{Q}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{Q}_{N-1}^{\mathrm{T}} \\ \frac{1-t_{K,N+2}}{p}\dot{\mathbf{Q}}_N^{\mathrm{T}} \\ \mathbf{Q}_N^{\mathrm{T}} \end{bmatrix} \tag{4.59}$$

und

$$\mathbf{B}_K = \begin{bmatrix} B_{0,p}\left(t'_0\right) & B_{1,p}\left(t'_0\right) & B_{2,p}\left(t'_0\right) & \cdots & B_{N,p}\left(t'_0\right) & B_{N+1,p}\left(t'_0\right) & B_{N+2,p}\left(t'_0\right) \\ -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ B_{0,p}\left(t'_1\right) & B_{1,p}\left(t'_1\right) & B_{2,p}\left(t'_1\right) & \cdots & B_{N,p}\left(t'_1\right) & B_{N+1,p}\left(t'_1\right) & B_{N+2,p}\left(t'_1\right) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ B_{0,p}\left(t'_{N-1}\right) & B_{1,p}\left(t'_{N-1}\right) & B_{2,p}\left(t'_{N-1}\right) & \cdots & B_{N,p}\left(t'_{N-1}\right) & B_{N+1,p}\left(t'_{N-1}\right) & B_{N+2,p}\left(t'_{N-1}\right) \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \\ B_{0,p}\left(t'_N\right) & B_{1,p}\left(t'_N\right) & B_{2,p}\left(t'_N\right) & \cdots & B_{N,p}\left(t'_N\right) & B_{N+1,p}\left(t'_N\right) & B_{N+2,p}\left(t'_N\right) \end{bmatrix} \tag{4.60}$$

formulieren, woraus die Kontrollpunktvektoren $\mathbf{P}_i, i \in \{0, \ldots, N+2\}$ über

$$\mathbf{P}_K = \mathbf{B}_K^{-1}\mathbf{Q}_K \tag{4.61}$$

berechnet werden. Die interpolierten Zustände $\mathbf{q}$ und $\dot{\mathbf{q}}$ sowie die interpolierten Eingänge $\mathbf{u}$ werden schließlich über

$$\begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \\ \mathbf{u}(t) \end{bmatrix} = \mathbf{C}_{NU}\left(t' = \frac{t}{T}\right) = \sum_{i=0}^{N+2} B_{i,p}\left(t'\right)\mathbf{P}_i \tag{4.62}$$

berechnet. Die Zustandsgrößen $\mathbf{x}_i(t)$ und Eingangsgrößen $\mathbf{u}_i(t)$ für $i \in \{1,2,3,h\}$ berechnen sich somit in jedem Zeitabschnitt $t \in [0,T_1]$, $t \in [T_1,T_1+T_2]$ und $t \in [T_1+T_2,T_1+T_2+T_3]$ aus einer eigenen B-Spline-Kurve $\mathbf{C}_{NU,i}(t')$, $i \in \{1,2,3\}$ des Roboters sowie $\mathbf{C}_{NU,h}(t')$ der Hand im Zeitabschnitt $t \in [T_1,T_1+T_2]$, gemäß

$$\begin{bmatrix}\mathbf{q}(t)\\ \dot{\mathbf{q}}(t)\\ \mathbf{u}(t)\end{bmatrix} = \begin{cases}\mathbf{C}_{NU,1}(t'=0), & \text{wenn } t<0\\ \mathbf{C}_{NU,1}\left(t'=\frac{t}{T_1}\right), & \text{wenn } 0 \le t < T_1\\ \mathbf{C}_{NU,2}\left(t'=\frac{t-T_1}{T_2}\right), & \text{wenn } T_1 \le t < T_1+T_2\\ \mathbf{C}_{NU,3}\left(t'=\frac{t-T_1-T_2}{T_3}\right), & \text{wenn } T_1+T_2 \le t < T_1+T_2+T_3\\ \mathbf{C}_{NU,3}(t'=1), & \text{wenn } t \ge T_1+T_2+T_3\end{cases} \tag{4.63a}$$

$$\begin{bmatrix}\mathbf{q}_h(t)\\ \dot{\mathbf{q}}_h(t)\\ \mathbf{u}_h(t)\end{bmatrix} = \begin{cases}\left[\mathbf{x}_{h,S}^{\mathrm{T}},\mathbf{0}^{\mathrm{T}}\right]^{\mathrm{T}}, & \text{wenn } t<T_1\\ \mathbf{C}_{NU,h}\left(t'=\frac{t-T_1}{T_2}\right), & \text{wenn } T_1 \le t < T_1+T_2\\ \left[\mathbf{x}_{h,E}^{\mathrm{T}},\mathbf{0}^{\mathrm{T}}\right]^{\mathrm{T}}, & \text{wenn } t \ge T_1+T_2\end{cases} \tag{4.63b}$$

(4.63c)

4.5 Online Trajektorienupdate

Die Genauigkeit der Posenschätzung während der Roboterbewegung wird mit sinkender Entfernung zwischen dem Objekt und der Kamera zunehmend besser. Wegen der sich ständig ändernden geschätzten Pose des Objekts ist ein Update der ursprünglich geplanten Trajektorie erforderlich. Da das Planen der Trajektorie über (4.44), (4.30) und (4.46) mehrere Sekunden benötigt, ist diese Methode für ein Trajektorienupdate während der Bewegung nicht geeignet. Auf Basis von [10, 11] wird nicht die gesamte Trajektorie neu geplant. Die Planung erfolgt für die Abweichung $\delta\boldsymbol{\xi}_i = [N\delta\Delta t, \delta\mathbf{x}_0, \ldots, \delta\mathbf{x}_N, \delta\mathbf{u}_0, \ldots, \delta\mathbf{u}_N]$, $i \in \{1,2,3\}$ für jeden Teilabschnitt $i \in \{1,2,3\}$ unter der Annahme, dass sich die geschätzte Pose des Objektes zwischen den einzelnen Updates nur gering ändert. Anschließend wird für die upgedateten Trajektorien $\boldsymbol{\xi}_i = \boldsymbol{\xi}_i^* + \delta\boldsymbol{\xi}_i$, $i \in \{1,2,3\}$ erneut eine Interpolation nach Kapitel 4.4 durchgeführt.

Ausgangspunkt für die Formulierung zur Berechnung der Updates ist die Trapez-Methode (4.2). Einsetzen von $\mathbf{x}_k = \mathbf{x}_k^* + \delta\mathbf{x}_k$, $\mathbf{u}_k = \mathbf{u}_k^* + \delta\mathbf{u}_k$ und $\Delta t = \Delta t^* + \delta\Delta t$ ergibt

$$\mathbf{x}_{k+1}^* + \delta\mathbf{x}_{k+1} - \mathbf{x}_k^* - \delta\mathbf{x}_k = \frac{\Delta t^* + \delta\Delta t}{2}\left(\mathbf{f}\left(\mathbf{x}_k^* + \delta\mathbf{x}_k, \mathbf{u}_k^* + \delta\mathbf{u}_k\right) + \mathbf{f}\left(\mathbf{x}_{k+1}^* + \delta\mathbf{x}_{k+1}, \mathbf{u}_{k+1}^* + \delta\mathbf{u}_{k+1}\right)\right). \tag{4.64}$$

Die Schrittweite Δt^* ist die Zeitdauer zwischen den einzelnen Stützstellen der betrachteten Trajektorie $\boldsymbol{\xi}_i^*$, $i \in \{1,2,3\}$. Aus der Taylor-Formel und Vernachlässigen der Terme zweiter und höherer Ordnung von $\delta\mathbf{x}_k$ und $\delta\mathbf{u}_k$ ergibt sich die linearisierte Systemdynamik zu

$$\mathbf{f}\left(\mathbf{x}_k^* + \delta\mathbf{x}_k, \mathbf{u}_k^* + \delta\mathbf{u}_k\right) = \mathbf{f}\left(\mathbf{x}_k^*, \mathbf{u}_k^*\right) + \left.\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_k^*,\mathbf{u}=\mathbf{u}_k^*}\delta\mathbf{x}_k + \left.\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\right|_{\mathbf{x}=\mathbf{x}_k^*,\mathbf{u}=\mathbf{u}_k^*}\delta\mathbf{u}_k. \tag{4.65}$$

Einsetzen der Taylor-Formel für $\mathbf{f}\left(\mathbf{x}_k^* + \delta\mathbf{x}_k, \mathbf{u}_k^* + \delta\mathbf{u}_k\right)$ bzw. $\mathbf{f}\left(\mathbf{x}_{k+1}^* + \delta\mathbf{x}_{k+1}, \mathbf{u}_{k+1}^* + \delta\mathbf{u}_{k+1}\right)$ in (4.64) und vernachlässigen der Terme mit $\delta\Delta t\delta\mathbf{x}_i, i \in \{k,k+1\}$ und $\delta\Delta t\delta\mathbf{u}_i, i \in \{k,k+1\}$ ergibt

$$\delta\mathbf{x}_{k+1} = \delta\mathbf{x}_k + \frac{\Delta t^*}{2}\left(\mathbf{\Gamma}_k^{\mathbf{x}}\delta\mathbf{x}_k + \mathbf{\Gamma}_k^{\mathbf{u}}\delta\mathbf{u}_k + \mathbf{\Gamma}_{k+1}^{\mathbf{x}}\delta\mathbf{x}_{k+1} + \mathbf{\Gamma}_{k+1}^{\mathbf{u}}\delta\mathbf{u}_{k+1}\right) + \frac{\delta\Delta t}{2}\left(\mathbf{f}_k^* + \mathbf{f}_{k+1}^*\right) \tag{4.66}$$

mit

$$\mathbf{f}_k^* = \mathbf{f}\left(\mathbf{x}_k^*, \mathbf{u}_k^*\right), \tag{4.67a}$$

$$\mathbf{\Gamma}_k^{\mathbf{x}} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_k^*,\mathbf{u}=\mathbf{u}_k^*} = \begin{bmatrix} \mathbf{0}_{7\times 7} & \mathbf{E}_{7\times 7} \\ \mathbf{0}_{7\times 7} & \mathbf{0}_{7\times 7} \end{bmatrix} \tag{4.67b}$$

$$\mathbf{\Gamma}_k^{\mathbf{u}} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\mathbf{x}=\mathbf{x}_k^*,\mathbf{u}=\mathbf{u}_k^*} = \begin{bmatrix} \mathbf{0}_{7\times 7} \\ \mathbf{E}_{7\times 7} \end{bmatrix} \tag{4.67c}$$

für $k = 0, \ldots, N-1$. N bzw. N_G ist dabei die Anzahl der Stützstellen der jeweils betrachteten Teiltrajektorie.

Die Abweichungen $\delta\boldsymbol{\xi}_i = \delta\boldsymbol{\xi}, i \in \{1,2,3\}$ jeder Teiltrajektorie berechnen sich jeweils aus einem Optimierungsproblem der Form

$$\min_{\delta\boldsymbol{\xi}} \quad J\left(\delta\boldsymbol{\xi}\right) = \frac{1}{2}\delta\boldsymbol{\xi}^{\mathrm{T}}\mathbf{Q}\delta\boldsymbol{\xi} \tag{4.68a}$$

$$\text{u.B.v.} \quad (4.66) \tag{4.68b}$$

$$\delta\mathbf{x}_0 = \overline{\mathbf{x}}_S - \mathbf{x}_0^*, \quad \delta\mathbf{x}_N = \overline{\mathbf{x}}_T - \mathbf{x}_N^* \tag{4.68c}$$

$$\underline{\delta\mathbf{x}} \le \delta\mathbf{x}_k \le \overline{\delta\mathbf{x}}, \quad k = 0, \ldots, N \tag{4.68d}$$

$$\underline{\delta\Delta t} \le \delta\Delta t \le \overline{\delta\Delta t} \tag{4.68e}$$

$$\underline{\boldsymbol{\tau}} - \underline{\mathbf{C}} \le \mathbf{D}\left(\mathbf{q}_k^* + \delta\mathbf{q}_k\right)\left(\mathbf{u}_k^* + \delta\mathbf{u}_k\right) + \mathbf{g}\left(\mathbf{q}_k^* + \delta\mathbf{q}_k\right) \le \overline{\boldsymbol{\tau}} - \overline{\mathbf{C}}, \quad k = 0, \ldots, N \tag{4.68f}$$

mit den neuen Startzuständen $\overline{\mathbf{x}}_S$, den neuen Endzuständen $\overline{\mathbf{x}}_T$ und der positiv definiten Gewichtungsmatrix $\mathbf{Q} = \operatorname{diag}\left(Q_{\Delta t}, \mathbf{Q}_{\mathbf{x}_0}, \ldots, \mathbf{Q}_{\mathbf{x}_N}, \mathbf{Q}_{\mathbf{u}_0}, \ldots, \mathbf{Q}_{\mathbf{u}_N}\right)$. Die unteren und oberen Grenzen sind über $\underline{\delta\mathbf{x}} = \underline{\mathbf{x}} - \mathbf{x}^*$ und $\overline{\delta\mathbf{x}} = \overline{\mathbf{x}} - \mathbf{x}^*$ gegeben. Durch die Grenzen in (4.68e) kann die maximale Änderung der Trajektoriendauer vorgegeben werden. Durch Versuche wurden sie zu $\underline{\delta\Delta t} = 0{,}0\,\mathrm{s}$ und $\overline{\delta\Delta t} = 0{,}01\,\mathrm{s}$ gewählt. Damit wird nur eine Verlängerung der Trajektoriendauer zugelassen. Durch die Beschränkungen (4.68f) bleibt die Summe aus dem bereits berechneten Momenten $\mathbf{u}_k^*$ und den optimalen Abweichungen $\delta\mathbf{u}$ im zulässigen Bereich, wodurch die Beschränkungen für diese auch hier berücksichtigt werden.

Für die erste Teiltrajektorie $C_{NU,1}\left(t' = \frac{t}{T_1}\right)$ wird zyklisch in einer Schleife ein Optimierungsproblem der Form (4.68) gelöst. Das Update wird dabei nicht für den gesamten Zeitbereich $t \in [0,T_1]$ berechnet, sondern vom Startzeitpunkt des jeweiligen Schleifdurchlaufs t_{act} inklusive eines genügend großen Zeitoffsets $\Delta t'$ bis zum Ende T_1 der Trajektorie. Der Zeitbereich, für den ein Update durchgeführt wird, ist damit $t \in [t_{act} + \Delta t', T_1]$. Die Anfangszustände sind über

$$\overline{\mathbf{x}}_S = C_{NU,1}\left(t' = \frac{t_{act} + \Delta t'}{T_1}\right) \tag{4.69}$$

gegeben. Die neuen Gelenkswinkel $\mathbf{q}_{T_1}$ zum Endzeitpunkt ergeben sich mit dem Update der Pose $\mathbf{H}_0^o$ des Objektes und (4.24) durch die Berechnung der inversen Roboterkinematik [21]. Diese wird numerisch mittels Gradientenmethode gelöst, wobei $C_{NU,1}(t'=1)_{[1:7]}$ als Startpunkt dient. Die Geschwindigkeiten $\dot{\mathbf{q}}$ werden unverändert aus der Interpolation $C_{NU,1}(t'=1)$ übernommen, womit der neue Zielzustand $\overline{\mathbf{x}}_T$ festgelegt ist. Da sich der betrachtete Zeithorizont $t \in [t_{act}+\Delta t', T_1]$ zunehmend verkleinert, müssen die optimalen Werte $\mathbf{x}_k^*$ sowie $\mathbf{u}_k^*$ für den Zeithorizont $t \in [t_{act}+\Delta t', T_1]$ in N äquidistanten Zeitpunkten durch (4.63a) neu berechnet werden. Über eine weitere temporäre Interpolation der Lösung $\delta\boldsymbol{\xi}^*$ aus (4.68) werden die einzelnen Werte der Abweichungen $\delta\boldsymbol{\xi}_1^*$ im Zeitbereich $t \in [t_{act}+\Delta t', T_1]$ berechnet. Die Werte in $\delta\boldsymbol{\xi}_1^*$ für den Zeitbereich $t \in [0, t_{act}+\Delta t']$ werden gleich Null gesetzt. Mit $\delta\boldsymbol{\xi}_1^*$ wird schließlich die erste Trajektorie $C_{NU,1}(t')$ gemäß Abschnitt 4.4 neu interpoliert und somit ein Update durchgeführt.

Zur Veranschaulichung dieser Vorgehensweise sind in Abbildung 4.7 die initiale Trajektorie und die ersten drei upgedateten Trajektorien des Gelenkswinkels $q_{1,i}(t), i = 0,1,\ldots$ mit der Trajektoriendauer $T_1 = 4\,\mathrm{s}$ dargestellt. Die initiale Trajektorie $q_{1,0}$ entspricht der Lösung des Optimierungsproblems (4.44), welches vorab offline berechnet wird. Der Zeitoffset wird auf den konstanten Wert $\Delta t' = 0{,}5\,\mathrm{s}$ gesetzt. Das Update der Trajektorie $q_{1,0}$ startet zum Zeitpunkt $t_{act} = 0{,}5\,\mathrm{s}$. Somit wird für den Zeitbereich $t \in [1\,\mathrm{s}, 4\,\mathrm{s}]$ über (4.68) die optimale Abweichung berechnet. Aus der initialen Trajektorie $q_{1,0}$ und der soeben berechneten optimalen Abweichung resultiert die Trajektorie $q_{1,1}$. Zum Zeitpunkt $t_{act} = 1{,}5\,\mathrm{s}$ startet die Berechnung des Updates der Trajektorie $q_{1,1}$ für den Zeitbereich $t \in [2\,\mathrm{s}, 4\,\mathrm{s}]$ und führt auf die Trajektorie $q_{1,2}$. Über ein weiteres Update zum Zeitpunkt $t_{act} = 2{,}5\,\mathrm{s}$ ergibt sich im Zeitbereich $t \in [3\,\mathrm{s}, 4\,\mathrm{s}]$ aus der Trajektorie $q_{1,2}$ die Trajektorie $q_{1,3}$. Wie gezeigt, werden die Zeitintervalle $t \in [t_{act}+\Delta t', T_1]$ der einzelnen Trajektorienupdates mit steigendem Index i kürzer.

Für die zweite Teiltrajektorie ändert sich der Anfangszustand, sowie durch die Bewegung der Finger und der Schwenkbewegung des Roboters auch der Endzustand. Aus dem neuen Endzustand der ersten Teiltrajektorie ergibt sich der Anfangszustand $\overline{\mathbf{x}_S}$ und über (4.24) der neue Endzustand $\overline{\mathbf{x}_T}$ der zweiten Teiltrajektorie. Die Geschwindigkeiten $\dot{\mathbf{q}}$ werden unverändert aus den Interpolationen $C_{NU,2}(t'=0)$ bzw. $C_{NU,2}(t'=1)$ übernommen. Mit der optimalen Abweichung $\delta\boldsymbol{\xi}_2^* = \delta\boldsymbol{\xi}^*$ wird für die zweite Teiltrajektorie $C_{NU,2}(t')$ ein Update durchgeführt. Das Update der zweiten Teiltrajektorie wird nur einmalig kurz vor dem Ende der ersten Teiltrajektorie berechnet.

Zuletzt wird ein einmaliges Update der dritten Teiltrajektorie $C_{NU,3}(t')$ mit $\delta\boldsymbol{\xi}_3^*$ aus (4.68) durchgeführt. Der Anfangszustand $\overline{\mathbf{x}_S}$ ist der Endzustand $C_{NU,2}(t'=1)$ der upgedateten zweiten Teiltrajektorie. Der Endzustand bleibt mit $\overline{\mathbf{x}_N} = C_{NU,3}(t'=1)$ unverändert.

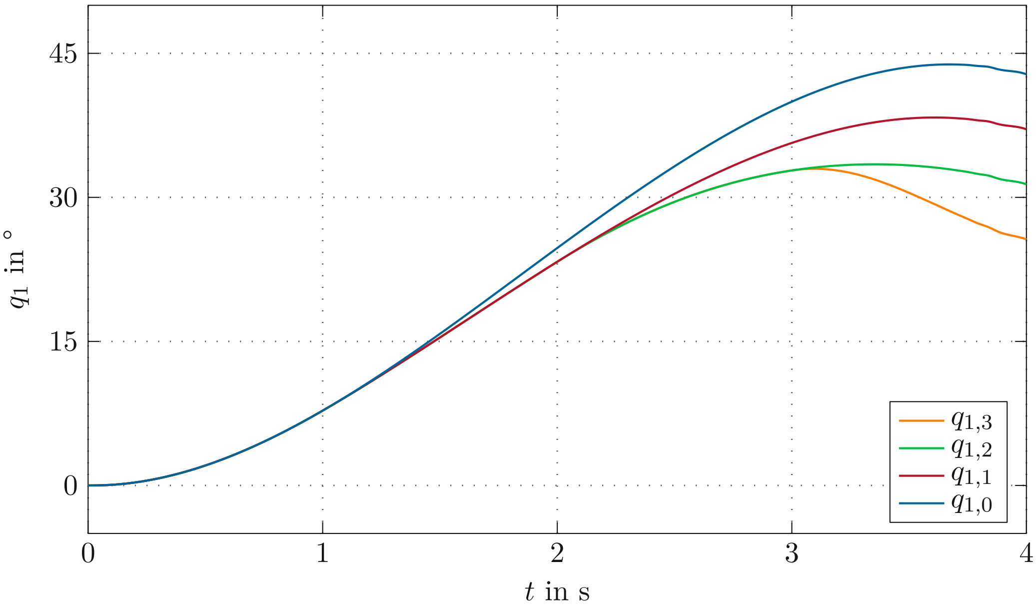


Abbildung 4.7: Exemplarische Darstellung der Initialtrajektorie $q_{1,0}$ und der Trajektorienupdates von $q_{1,i}, i \in \{1,2,3\}$.

5 Trajektorienfolgeregler

Die geplanten Trajektorien aus dem vorigen Kapitel werden vorgesteuert. Um etwaige Abweichungen gering zu halten bzw. zu eliminieren, wird in diesem Kapitel für den Roboter und die Finger ein Trajektorienfolgeregler entworfen. In Abschnitt 5.1 wird für den Roboter ein Computed-Torque Regler mit Integralanteil vorgestellt. Zusätzlich hat der Roboter unterlagert eine Regelung basierend auf der singulären Störtheorie und eine Reibungskompensation. Der Abschnitt 5.2 beschäftigt sich mit dem Entwurf des Trajektorienfolgereglers der Fingergelenke. Aufgrund der internen Steuerung bzw. Regelung der Greifhand wird der Trajektorienfolgeregler auf Basis einer Zwei-Freiheitsgrad-Regelkreisstruktur entworfen. Das Führungssignal für den internen Regler der Greifhand ist die gewünschte Geschwindigkeit $\dot{\mathbf{q}}_{h,d}$ der Fingergelenke. Durch die Wahl einer Zwei-Freiheitsgrad-Regelkreisstruktur werden etwaige Abweichungen zwischen den gewünschten $\mathbf{q}_{h,d}$ und den tatsächlichen Zuständen $\mathbf{q}_h$ berücksichtigt bzw. kompensiert.

5.1 Roboter

Für den Trajektorienfolgeregler des Roboters wird das sog. Regelgesetz der inversen Dynamik bzw. Computed-Torque mit Integralanteil [15]

$$\boldsymbol{\tau}_d = \mathbf{D}(\mathbf{q})\,\mathbf{v} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{5.1}$$

mit dem neuen Stelleingang $\mathbf{v}$ verwendet. Wie bereits in Kapitel 4 erläutert, ist der Einfluss der Coriolis-Matrix gering und wird dadurch vernachlässigt. Das vorgegebene Drehmoment errechnet sich demnach durch

$$\boldsymbol{\tau}_d = \mathbf{D}(\mathbf{q})\,\mathbf{v} + \mathbf{g}(\mathbf{q})\,, \tag{5.2}$$

wobei mit $\mathbf{v}$ der neue Stelleingang bezeichnet wird. Der Stelleingang $\mathbf{v}$ ermöglicht nun die Formulierung des Trajektorienfolgereglers in der Form

$$\mathbf{v} = \ddot{\mathbf{q}}_d - \mathbf{K}_1(\dot{\mathbf{q}} - \dot{\mathbf{q}}_d) - \mathbf{K}_0(\mathbf{q} - \mathbf{q}_d) - \mathbf{K}_I \int \mathbf{q} - \mathbf{q}_d \mathrm{d}t \tag{5.3}$$

mit der gewünschten Trajektorie $\mathbf{q}_d$, $\dot{\mathbf{q}}_d$ und $\ddot{\mathbf{q}}_d(t) = \mathbf{u}(t)$ aus (4.63a), und den positiv definiten Diagonalmatrizen $\mathbf{K}_0$, $\mathbf{K}_1$ und $\mathbf{K}_I$. Es kann einfach gezeigt werden, dass der geschlossene Kreis asymptotisch stabil ist.

5.2 Greifhand

Im folgenden Abschnitt wird ein Trajektorienfolgeregler für die Fingergelenke der Greifhand entworfen. An dieser Stelle sei erwähnt, dass es durch die interne Steuerung der Hand

nicht möglich ist, direkt die Drehmomente der Fingergelenke vorzugeben. In der internen Steuerung der Greifhand wird einer der folgenden Regler festgelegt:

1. Ein Positionsregler für $\mathbf{q}_h$ mit konstanter maximaler Geschwindigkeit $|\dot{\mathbf{q}}_h| = \dot{\mathbf{q}}_{h,max}$ und konstanter maximaler Beschleunigung $|\ddot{\mathbf{q}}_h| = \ddot{\mathbf{q}}_{h,max}$.
2. Ein Geschwindigkeitsregler für $\dot{\mathbf{q}}_h$ mit einer konstanten internen Beschleunigung $|\ddot{\mathbf{q}}_h| = \ddot{\mathbf{q}}_{h,max}$.
3. Ein Geschwindigkeitsregler für $\dot{\mathbf{q}}_h$ mit vorgegebener Beschleunigung $|\ddot{\mathbf{q}}_h|$.

Nähere Angaben zur Konfiguration der Hand folgen in Kapitel 6.

Als interner Regler wird der Geschwindigkeitsregler für $\dot{\mathbf{q}}_h$ mit einer konstanten internen Beschleunigung $|\ddot{\mathbf{q}}_h| = \ddot{\mathbf{q}}_{h,max}$ gewählt. Da der interne Regler nur auf die gewünschte Geschwindigkeit $\dot{\mathbf{q}}_{h,d}$ regelt, können Abweichungen zwischen den gewünschten $\mathbf{q}_{h,d}$ und den tatsächlichen Zuständen $\mathbf{q}_h$ auftreten. Um dem entgegen zu wirken, wird eine Regelkreisstruktur wie in Abbildung 5.1 aufgebaut. Der Sollwert für den internen Geschwindigkeitsregler $\dot{\mathbf{q}}_{h,s}$ bildet sich aus der Summe des Steuerungsanteils $\dot{\mathbf{q}}_{h,d}$ und des Regleranteils $\dot{\mathbf{q}}_{h,e}$ eines PI-Reglers, entsprechend

$$\dot{\mathbf{q}}_{h,s} = \dot{\mathbf{q}}_{h,d} + \dot{\mathbf{q}}_{h,e}. \tag{5.4}$$

Für den PI-Regler gilt

$$\dot{\mathbf{q}}_{h,e} = \mathbf{K}_P \mathbf{q}_{h,e} + \mathbf{K}_I \int \mathbf{q}_{h,e} \mathrm{d}t, \qquad \mathbf{q}_{h,e} = \mathbf{q}_{h,d} - \mathbf{q}_h \tag{5.5}$$

mit den Diagonalmatrizen $\mathbf{K}_P$ und $\mathbf{K}_I$. Die gewählten Werte für $\mathbf{K}_P$ und $\mathbf{K}_I$ sind der Tabelle A.7 zu entnehmen.

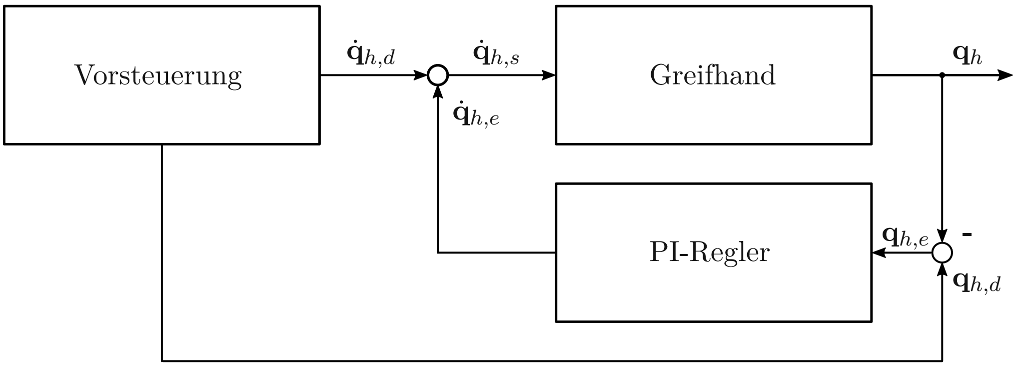


Abbildung 5.1: Blockschaltbild der Regelkreisstruktur der Greifhand.

6 Implementierung und Ergebnisse

Der erste Teil dieses Kapitels beinhaltet eine genauere Beschreibung des Versuchsaufbaus und wie die einzelnen Komponenten miteinander kommunizieren. Im zweiten Teil ist eine offline geplante Trajektorie für eine zufällige Objektpose dargestellt. Zusätzlich wird für vier zufällige Objektposen ein Greifen ausgeführt. In Abschnitt 6.3 wird eine Monte-Carlo-Simulation der Offline-Trajektorienplanung durchgeführt und die Erfolgsrate beurteilt.

6.1 Implementierung

Dieses Kapitel beschreibt die Implementierung des Versuchs und geht auf die erzielten Ergebnisse ein. Der Versuchsaufbau ist in Abbildung 6.1 dargestellt. Der KUKA Roboter ist über EtherCAT mit dem PC verbunden. Die Regelung des Roboters erfolgt über das vom ACIN zur Verfügung gestellte MATLAB/SIMULINK Modell. Durch das Beckhoff TE1400 TC3 Target for Simulink® wird aus dem Modell ein Modul generiert, das im TwinCAT geladen wird und in der Echtzeitumgebung von TwinCAT mit einer Zykluszeit von 125 µs läuft. Das zur Verfügung gestellte MATLAB/SIMULINK Modell wurde um das Kalman-Filter aus Abschnitt 3.2 und die Berechnung der Trajektorie (4.63a) erweitert. Der Trajektorienfolgeregler aus Abschnitt 5.1 ist bereits Teil des ACIN-Modells. Des Weiteren ist unterlagert noch eine Regelung basierend auf der singulären Störtheorie sowie eine Reibungskompensation. Die Laufzeit t, mit welcher die Solltrajektorie berechnet wird, wird manuell vom Zeitpunkt $t = 0\,\mathrm{s}$ gestartet.

Die Schunk Greifhand ist über eine weitere Netzwerkkarte mit dem PC verbunden. Für die Steuerung und Regelung nach Abschnitt 5.2 wurde ein weiteres MATLAB/SIMULINK Modell erstellt, welches unabhängig vom Roboter in einer eigenen TwinCAT RT Umgebung mit einer Zykluszeit von 20 ms läuft. Es wurde ein TwinCAT C/C++ Modul programmiert, welches für den zyklischen Datenaustausch der internen Steuerung der Greifhand und dem MATLAB/SIMULINK Modell über Real-Time-Ethernet zuständig ist. Im MATLAB/SIMULINK Modell kann manuell einer der zur Verfügung stehenden internen Regler der Greifhand (siehe Kapitel 5.2) vorgewählt werden. Zur Auswahl steht ein Positionsregler mit konstanter maximaler Geschwindigkeit $|\dot{\mathbf{q}}_h| = \dot{\mathbf{q}}_{h,max}$ und konstanter maximalen Beschleunigung $|\ddot{\mathbf{q}}_h| = \ddot{\mathbf{q}}_{h,max}$. Des Weiteren können zwei unterschiedliche Geschwindigkeitsregler $\dot{\mathbf{q}}_h$, mit der maximalen Beschleunigung $|\ddot{\mathbf{q}}_h| = \ddot{\mathbf{q}}_{h,max}$ bzw. mit vorgegebener Beschleunigung $|\dot{\mathbf{q}}_h|$ der Fingergelenke, ausgewählt werden. Je nach ausgewähltem Regler werden unterschiedliche Steuerkommandos an die Greifhand gesendet. Das erstellte C/C++ Modul sendet je nach Reglerauswahl die erforderlichen Befehle, liest die internen Zustände der Hand und leitet sie an das MATLAB/SIMULINK Modell weiter. Zusätzlich beinhaltet das MATLAB/SIMULINK Modell die Berechnung der Solltrajektorie (4.63b) und den Trajektorienfolgeregler aus Abschnitt 5.2. Die Laufzeit t des Versuchs

wird aus dem MATLAB/SIMULINK Modell des Roboters ausgelesen. Aus Gründen der Performance können die Steuerbefehle maximal mit 50 Hz an die Greifhand gesendet werden. Die Greifhand ist mit drucksensitiven Flächen ausgestattet, die in dieser Arbeit jedoch nicht ausgewertet wurden. Genauere Informationen über die möglichen Einstellungen sind dem Manual zu entnehmen [22].

Die Kamera ist über Gigabit mit dem PC verbunden. Für das Auslesen der Bilder wird das Python Modul pypylon verwendet [23], welches von Basler zur Verfügung gestellt wird. Die Bilder werden in einem Python Skript zyklisch ausgelesen. Für die Bildverarbeitung, das Erkennen der ArUco-Marker, das Extrahieren der Punktematrix $\bar{\mathbf{p}}$ und das Berechnen von (3.6) werden die jeweiligen Implementierungen der frei verfügbaren Bibliothek OpenCV Version 4.2 inklusive der Extramodule verwendet [24, 25]. In jedem Bild werden die ArUco-Marker detektiert, die Pose des Koordinatensystems $(0_o x_o y_o z_o)$ in Bezug zur Kamera $(0_c x_c y_c z_c)$ berechnet und über die ADS-Bibliothek [26] an das Kalman-Filter im Roboter-Modell weitergeleitet.

Das Online-Update aller Teiltrajektorien erfolgt über ein weiteres Python Skript. In dem Skript werden aus dem Robotermodell über ADS die aktuelle Zeit $t_{act} = t$ und die geschätzte Pose des Objekts in Bezug zur Kamera ausgelesen. Durch das Kalman-Filter wird die Roboterbewegung zwischen zwei Bildern berücksichtigt und somit ist eine genauere Posenschätzung möglich. Zusätzlich wird durch das Kalman-Filter der Einfluss von Ausreißer erheblich reduziert und es kann auf eine Detektion und Entfernung von Ausreißern verzichtet werden. Aus der geschätzten Pose und dem Robotermodell wird die Pose des Objekts in Bezug zum Basiskoordinatensystems berechnet und anschließend wie in Kapitel 4.5 beschrieben ein Update der ersten Teiltrajektorie durchgeführt. Durch das Update ergeben sich neue Kontrollpunktvektoren $\mathbf{P}_i, i \in \{i = 0{,}1, \ldots, N+2\}$ für (4.62). Diese Kontrollpunktvektoren werden über ADS an die Trajektorienberechnung im Robotermodell übergeben. Der Zeitoffset $\Delta t'$ in (4.69) wird so groß gewählt, dass das Trajektorienupdate innerhalb $\Delta t'$ abgeschlossen ist und somit mögliche Sprünge in der Solltrajektorie vermieden werden, auch wenn es sich nur um kleine Sprünge handelt. Dieses Update der ersten Teiltrajektorie wird im Zeitintervall $t \in [0; 0{,}9T_1]$ in einer Schleife ausgeführt. Gilt für die aktuelle Zeit $t_{act} > 0{,}9T_1$, wird einmalig ein Update der zweiten und der dritten Teiltrajektorie nach Abschnitt 4.5 durchgeführt und die neuen Kontrollpunktvektoren der Trajektorie an das Robotermodell übergeben.

Die Optimierungsprobleme aus Kapitel 4 wurden alle mit dem *Interior Point Optimizer* [27], kurz IPOPT, und dem Linear Solver MA57 [28, 29] gelöst. Die Ableitungen der Kostenfunktionen und der Beschränkungen wurden über das Open-Source Tool CasADi [30] realisiert, welches *automatic differentiation* (AD) ermöglicht.

6.2 Ergebnisse

Im folgendem Abschnitt werden einige Ergebnisse dargestellt und diskutiert. Ursprünglich war eine Objekterkennung mittels Künstlicher Intelligenz vorgesehen. Dazu wurden die Neuronale Netze aus der Singleshotpose-Methode [5], kurz SSP, und der PVnet-Methode [4] implementiert. Für das Training der Netzwerke wurde unter Zuhilfenahme des *NVIDIA Deep learning Dataset Synthesizer* [6] ein Datenset aus 10000 Bildern erstellt. Für eine

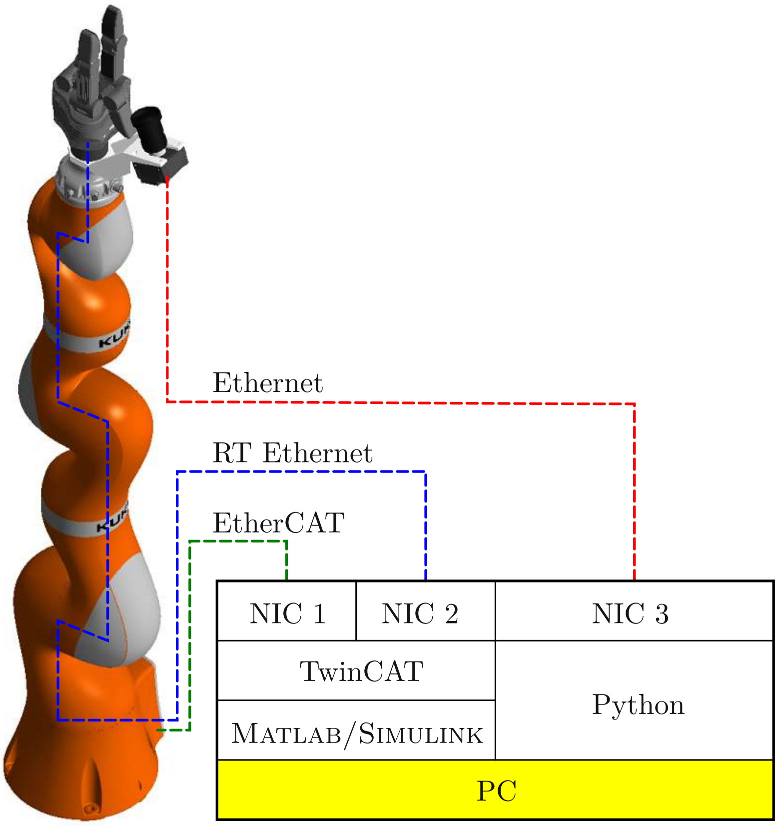


Abbildung 6.1: Übersicht des Versuchsaufbaus.

Bildauflösung von 1024×1024 Px hat das Netzwerk aus der PVnet-Methode eine Inferenzzeit von rund 200 ms und ist somit für eine Real-Time Anwendung weniger geeignet. Im Gegensatz dazu liegt die Inferenzzeit des Netzwerks der SSD-Methode bei rund 40 ms. Es hat sich bei beiden Varianten gezeigt, dass die Genauigkeit der geschätzten Pose stark von der Objektgröße im Bild abhängig ist. Mit zunehmender Größe steigt der Fehler zwischen der geschätzten und der tatsächlichen Pose. Ist die Entfernung zwischen Kamera und Objekt zu klein, scheitern beide Varianten. Aus diesem Grund wurde zu der in Kapitel 3 beschriebenen Methode mit ArUco-Markern gewechselt.

Zu Beginn des Versuchs befindet sich der Roboter in einen der beiden gewählten Initialzustände

$$\mathbf{q}_0^{\mathrm{T}} = [0°,0°,0°,-40°,0°,100°,90°]\,, \tag{6.1a}$$
$$\mathbf{q}_0^{\mathrm{T}} = [55°,39{,}5°,-6{,}6°,-22°,-36{,}3°,110{,}6°,134°]\,. \tag{6.1b}$$

Als Zielpunkt bzw. Endzustand wird

$$\mathbf{q}_T^{\mathrm{T}} = [0°,0°,0°,-40°,0°,100°,90°] \tag{6.2}$$

festgelegt. Das Objekt wird im Sichtfeld der Kamera am Boden platziert und anhand der Schätzung der Pose offline eine Trajektorie geplant. In den Abbildungen 6.2, 6.3 und 6.4 ist die über (4.63a) berechnete Trajektorie des Roboters mit dem Initialzustand (6.1b) im Zeitintervall $t \in [0,T_1+T_2+T_3]$ mit $T_1 = T_3 = 4\,\mathrm{s}$ und $T_2 = 0{,}5\,\mathrm{s}$ dargestellt. Der blaue Teil der Trajektorie ist die Roboterbewegung vom Initialzustand des Roboters bis zum Beginn des Greifens, der rote Teil zeigt die Roboterbewegung während des Schließens der Finger, und der grüne Teil zeigt die Bewegung zur Zielposition nach dem Greifen. Wie gewünscht gehen die einzelnen Teiltrajektorien stetig ineinander über. Die über (4.63b) berechnete Trajektorie der Fingergelenke im Zeitintervall $t \in [T_1,T_1+T_2]$ zeigt Abbildung 6.5. Finger 1 und 3 sind bis zum Zeitpunkt $t = T_1$ symmetrisch zueinander, somit gilt $q_{h2}(t <= T_1) = q_{h6}(t <= T_1)$ sowie $q_{h3}(t <= T_1) = q_{h7}(t <= T_1)$. Die geplanten Eingangsgrößen $u_{hi}, i \in \{2,3,6,7\}$ sind ebenfalls dargestellt, werden jedoch für die Regelung der Fingergelenke nicht benötigt. Sie wurden mitgeführt, um denselben Code für die Interpolation und dieselbe Struktur für die Trajektorienplanung verwenden zu können. Die Darstellung zeigt, dass mit der festgelegten Zeitdauer von $T_2 = 0{,}5\,\mathrm{s}$ fast die kürzeste realisierbare Zeit für das Greifen erreicht ist, da der Gelenkswinkel q_{h6} mit konstanter maximaler Beschleunigung beschleunigt und wieder abbremst. Die aus der Trajektorie der Gelenkswinkel $\mathbf{q}_h$ berechnete Trajektorie der Greifzustände $\mathbf{q}_G$ ist in Abbildung 6.6 dargestellt. Sie zeigt, dass der im Optimierungsproblem (4.19) geforderte lineare Anstieg von y_G und somit die Gleichungsbeschränkung (4.19c) eingehalten wird.

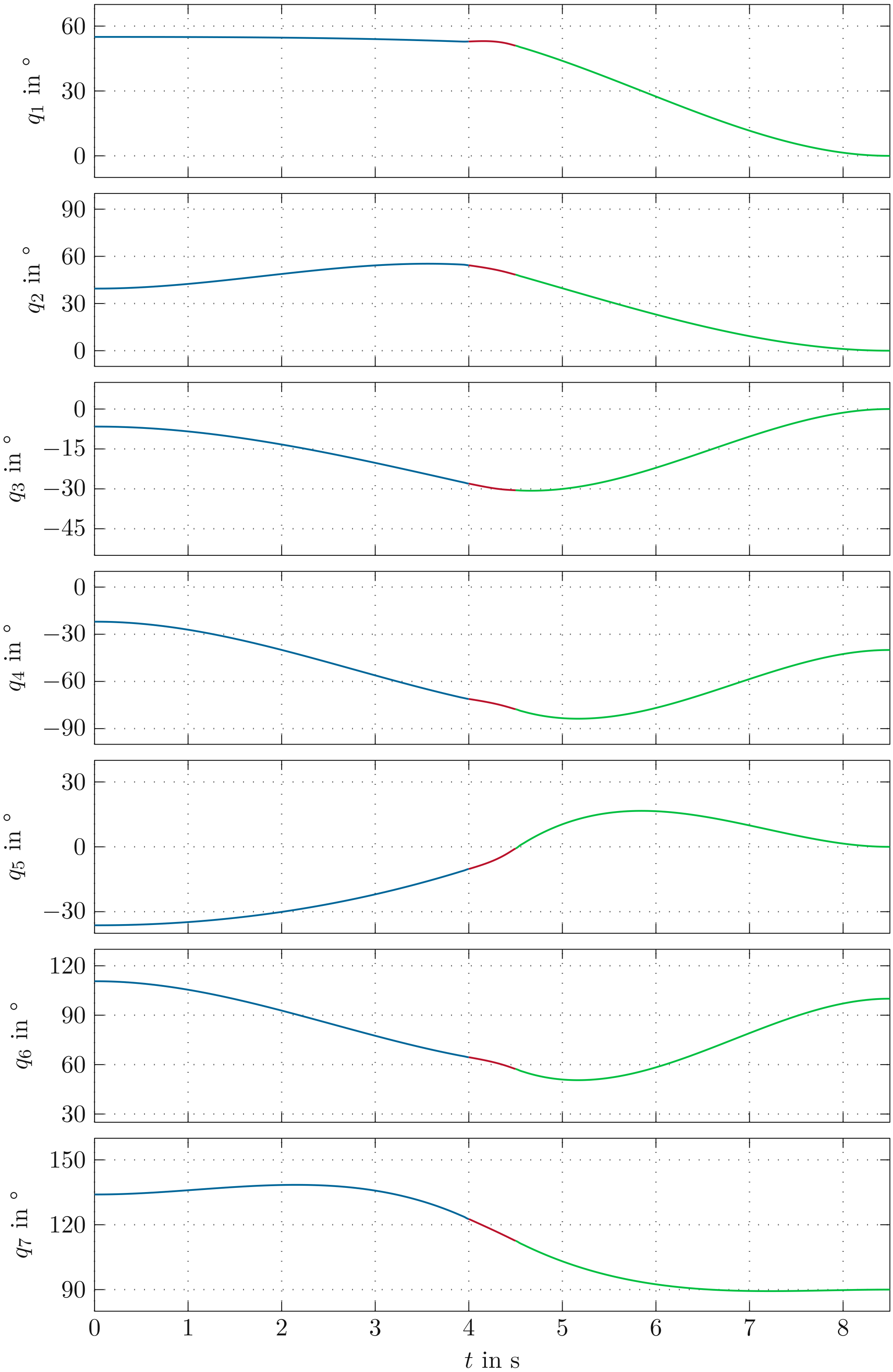


Abbildung 6.2: Interpolierte Teiltrajektorien der Gelenkswinkel $q_1, \ldots, q_7$ des Roboters aus (4.44) in blau, (4.30) in rot und (4.46) in grün.

6 Implementierung und Ergebnisse

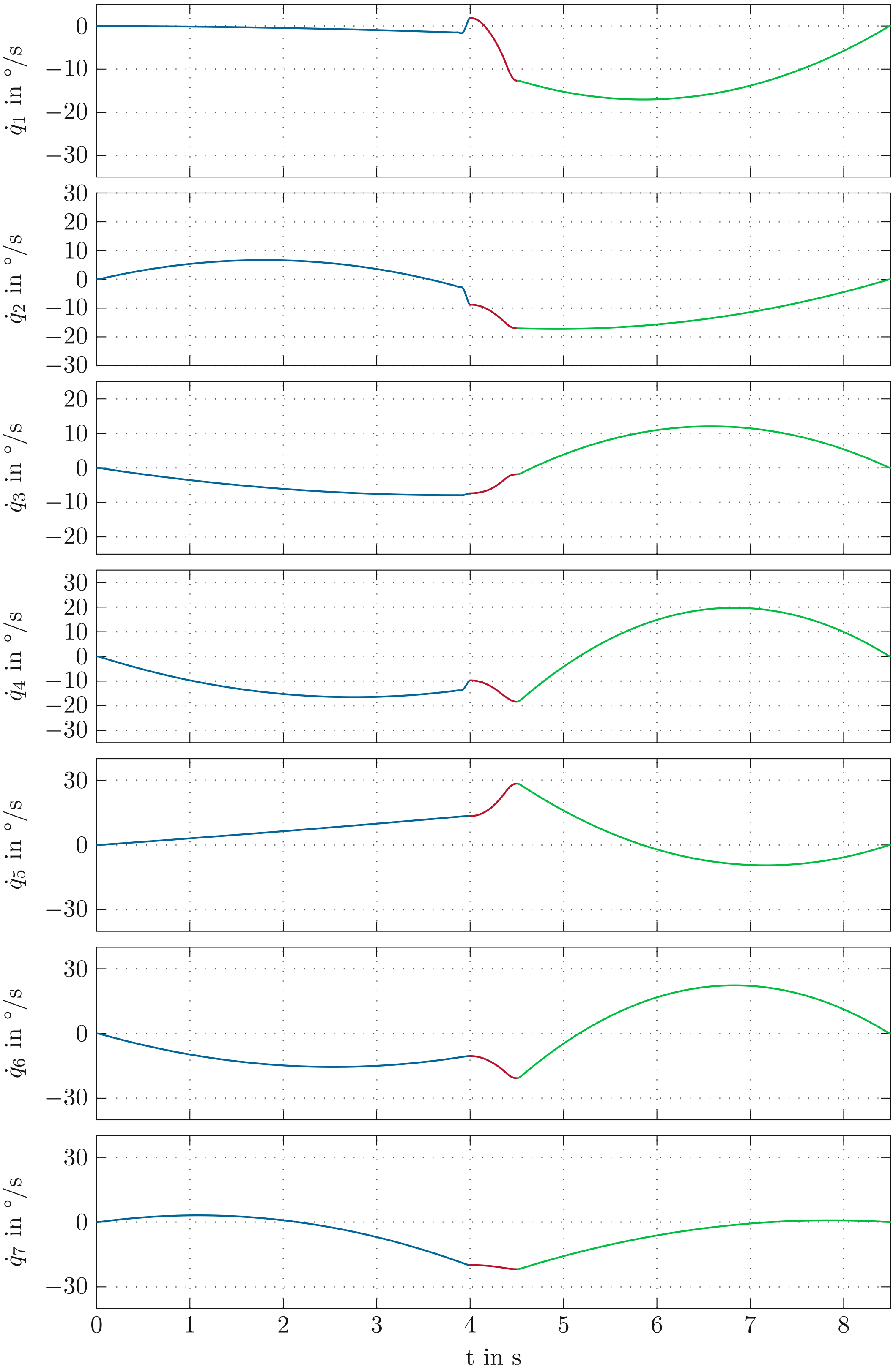


Abbildung 6.3: Interpolierte Teiltrajektorien der Gelenkswinkelgeschwindigkeiten $\dot{q}_1, \ldots, \dot{q}_7$ des Roboters aus (4.44) in blau, (4.30) in rot und (4.46) in grün.

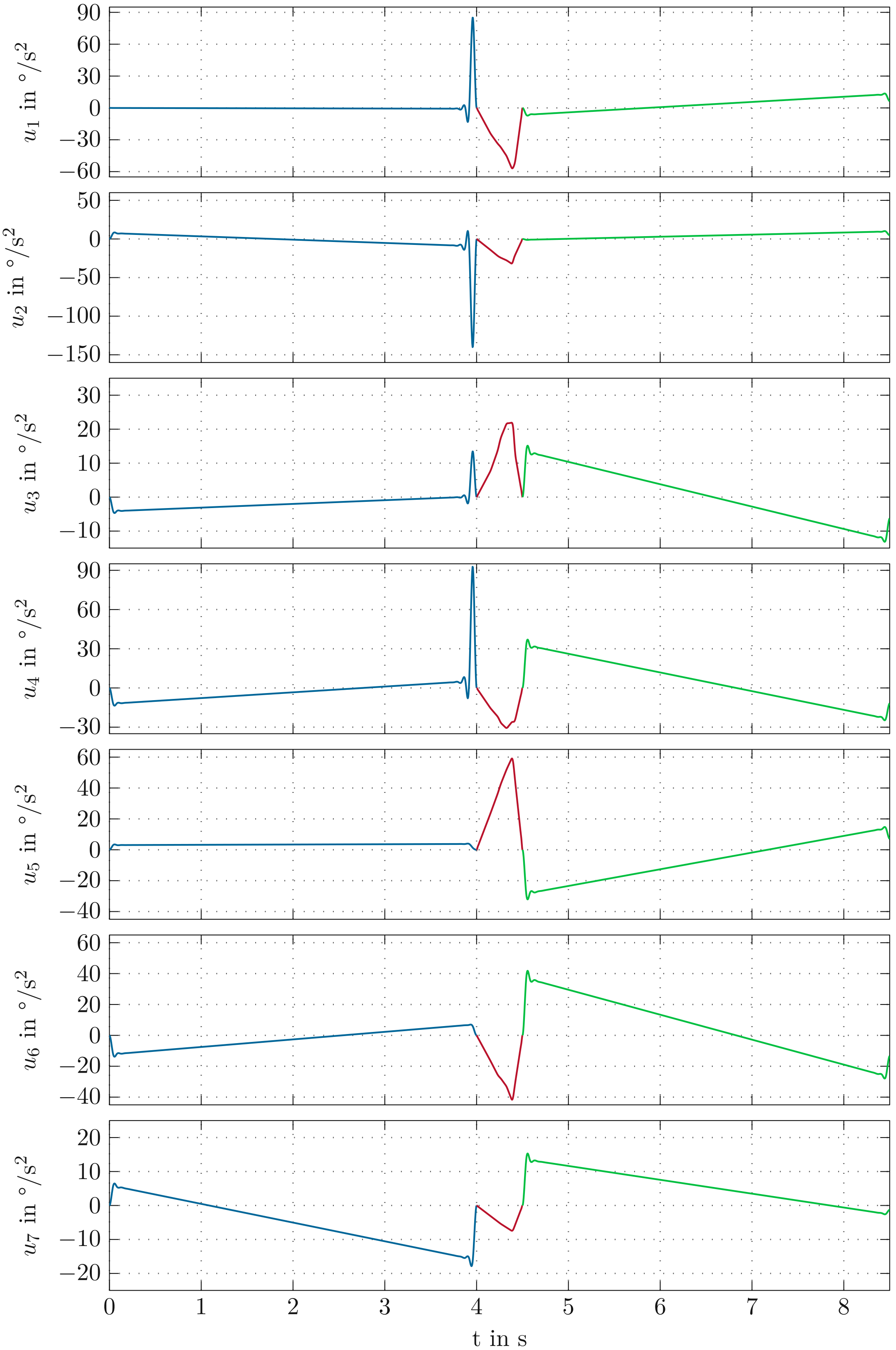


Abbildung 6.4: Interpolierte Teiltrajektorien der Gelenkswinkelbeschleunigungen $\ddot{q}_1, \ldots, \ddot{q}_7$ des Roboters aus (4.44) in blau, (4.30) in rot und (4.46) in grün.

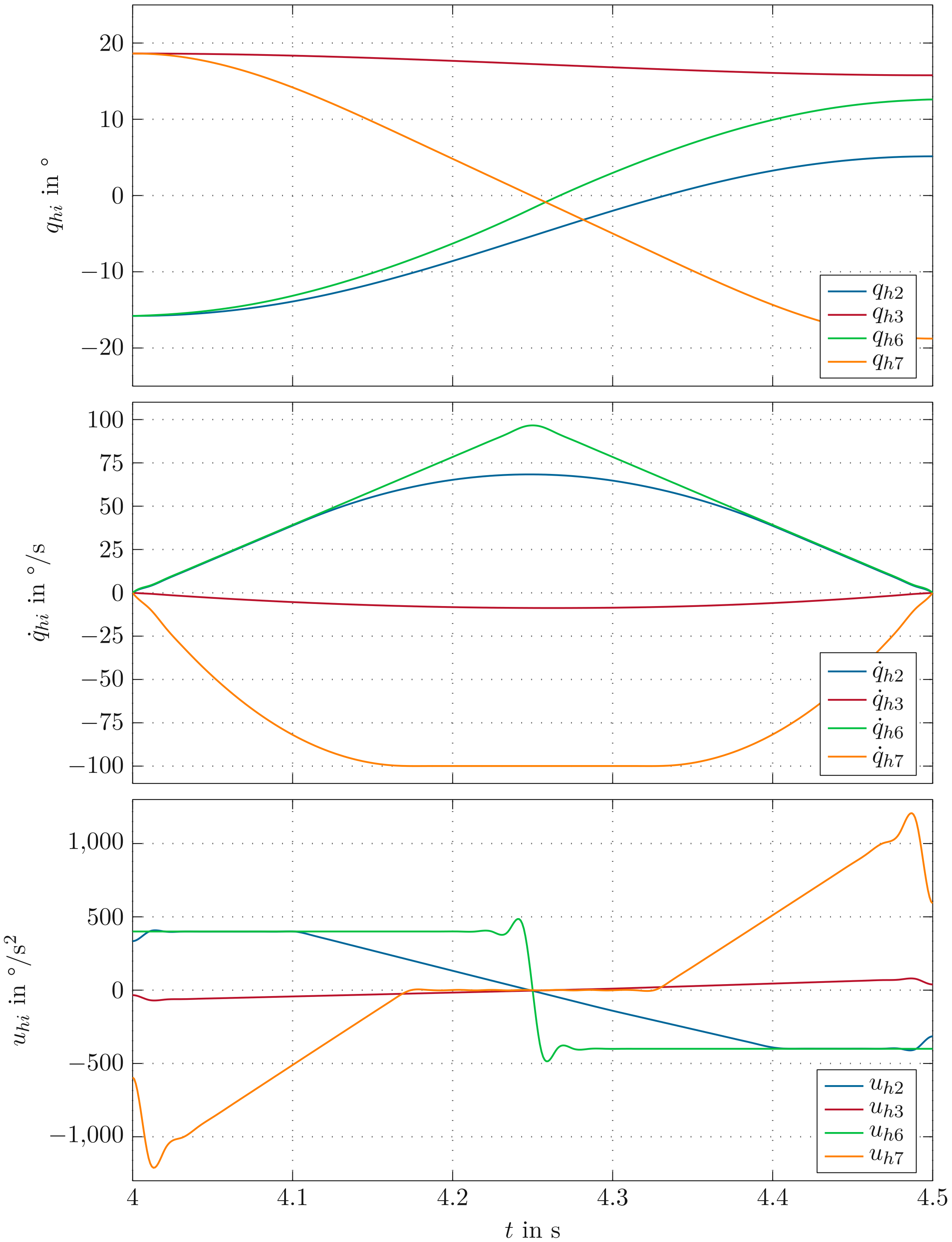


Abbildung 6.5: Interpolierte Teiltrajektorien der Fingergelenke aus (4.19).

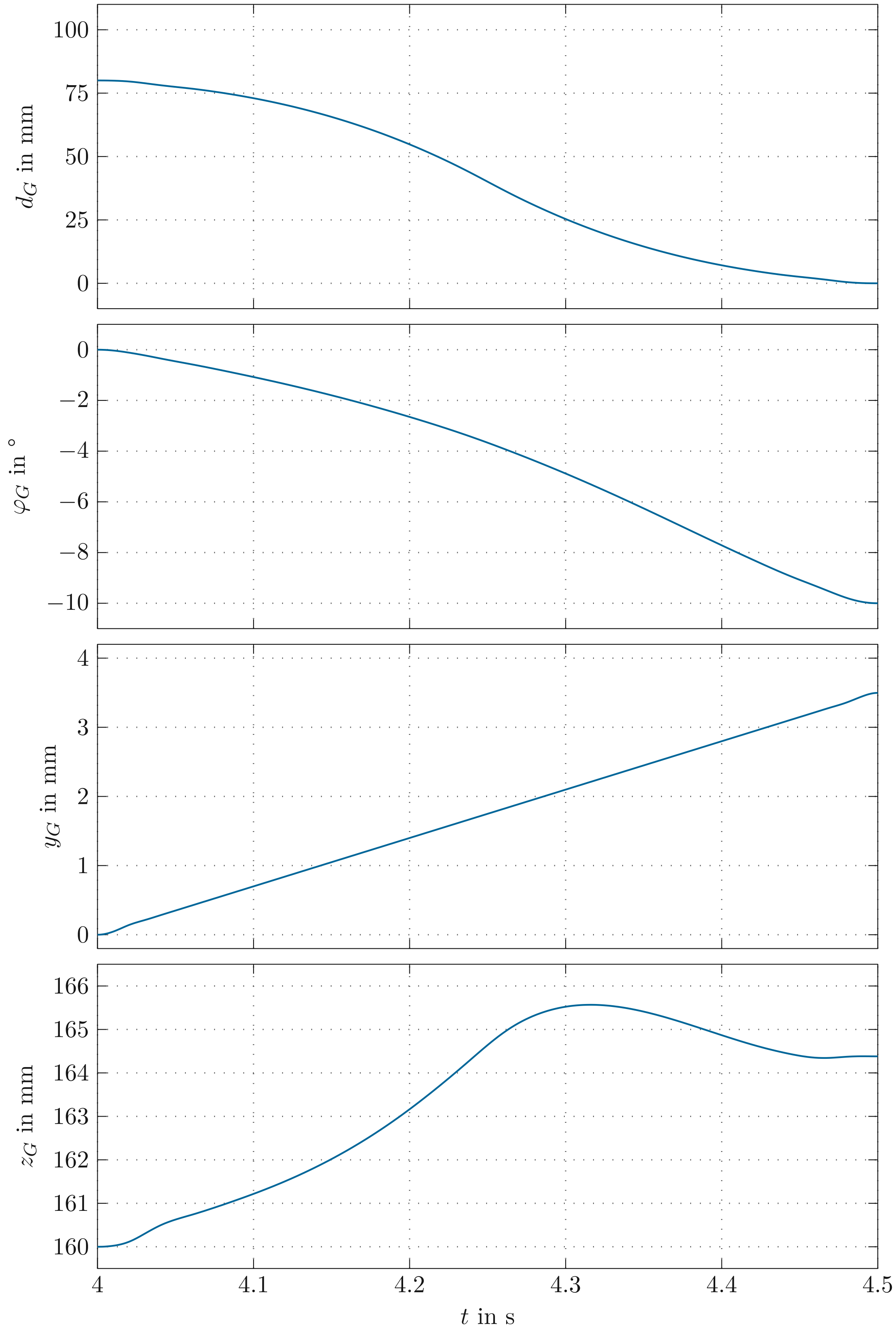


Abbildung 6.6: Greifzustände aus den interpolierten Teiltrajektorien der Fingergelenke aus (4.19).

Mit jedem Initialzustand in (6.1) wurden zwei Versuche mit einer zufälligen Objektpose durchgeführt, dargestellt in den Abbildungen 6.7-6.12. Die Trajektorien der Gelenksgrößen $\mathbf{q}$ sind in Abbildung 6.7 mit dem Initialzustand (6.1a) sowie in Abbildung 6.10 mit dem Initialzustand (6.1b) dargestellt. Die Trajektorien der Geschwindigkeiten $\dot{\mathbf{q}}$ sind in den Abbildungen 6.8 und 6.11 abgebildet, sowie die Trajektorien der Beschleunigungen in den Abbildungen 6.9 und 6.12. Die Abbildungen enthalten die geplanten Trajektorien aus der Offline-Trajektorienplanung in blau und orange und die upgedateten Trajektorien aus dem Online-Trajektorienupdate am Ende des Versuchs in rot und grün. Wie zu sehen ist, sind auch nach mehreren online Trajektorienupdates die Trajektorien stetig und es treten keine Sprünge auf. Es zeigt auch, dass der Roboter im Zeitbereich $t \in [0,T_1]$ mit $T_1 = 4\,\mathrm{s}$ hauptsächlich der ursprünglich geplanten optimalen Trajektorie folgt und erst im letzten Viertel $t \in [3\,\mathrm{s},4\,\mathrm{s}]$ größere Abweichungen auftreten.

In Abbildung 6.13 sind in blau jene Zeitpunkte eingezeichnet, an denen ein Messwert für das Kalman-Filter Messupdate zur Verfügung steht, sowie in rot die Zeitpunkte an denen ein online Update der Trajektorie ausgeführt wurde. Die Zeitdifferenz zwischen den einzelnen Zeitpunkten ist nicht konstant, da die Ausführungszeit für das Auswerten der Bilder und Berechnen der Pose nicht konstant ist und nicht in der Echtzeitumgebung von TwinCAT implementiert ist. Gleiches gilt auch für das online Update der Trajektorie. Abbildung 6.13 zeigt, dass während der Bewegung vom Initialzustand zum Objekt im Schnitt 30 Messupdates des Kalman-Filters und 17 Trajektorienupdates stattfinden. Die mittlere Zykluszeit für das Messupdate des Kalman-Filters ist 161,3 ms und 185 ms für das Trajektorienupdate.

Während der Versuche hat sich gezeigt, dass die numerische Berechnung der inversen Kinematik nicht immer konvergiert. In diesem Fall wird die Zielposition aus dem vorigen Trajektorienupdate übernommen.

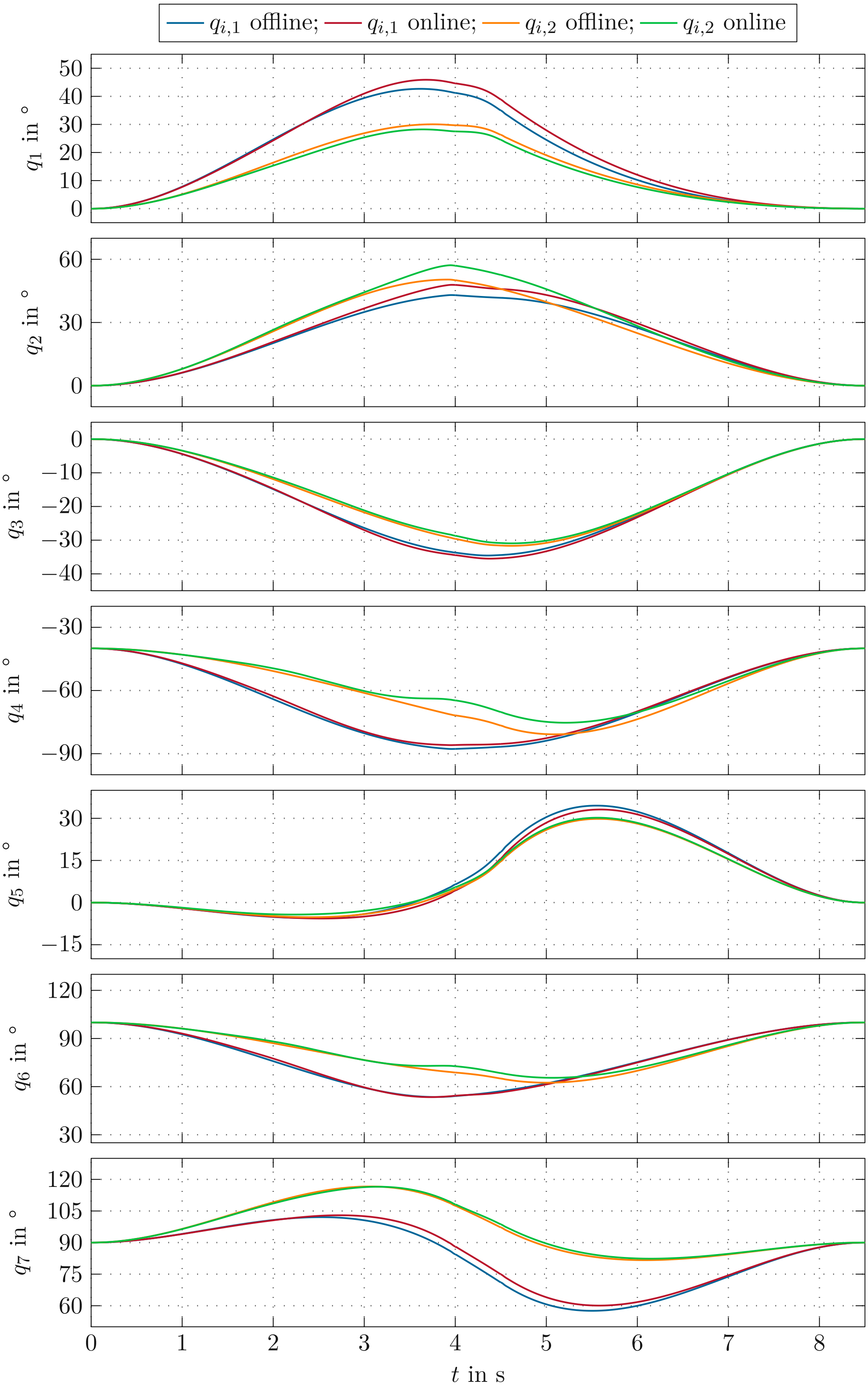


Abbildung 6.7: Interpolierte Teiltrajektorien der Gelenkswinkel $q_1, \ldots, q_7$ des Roboters mit Initialzustand (6.1a).

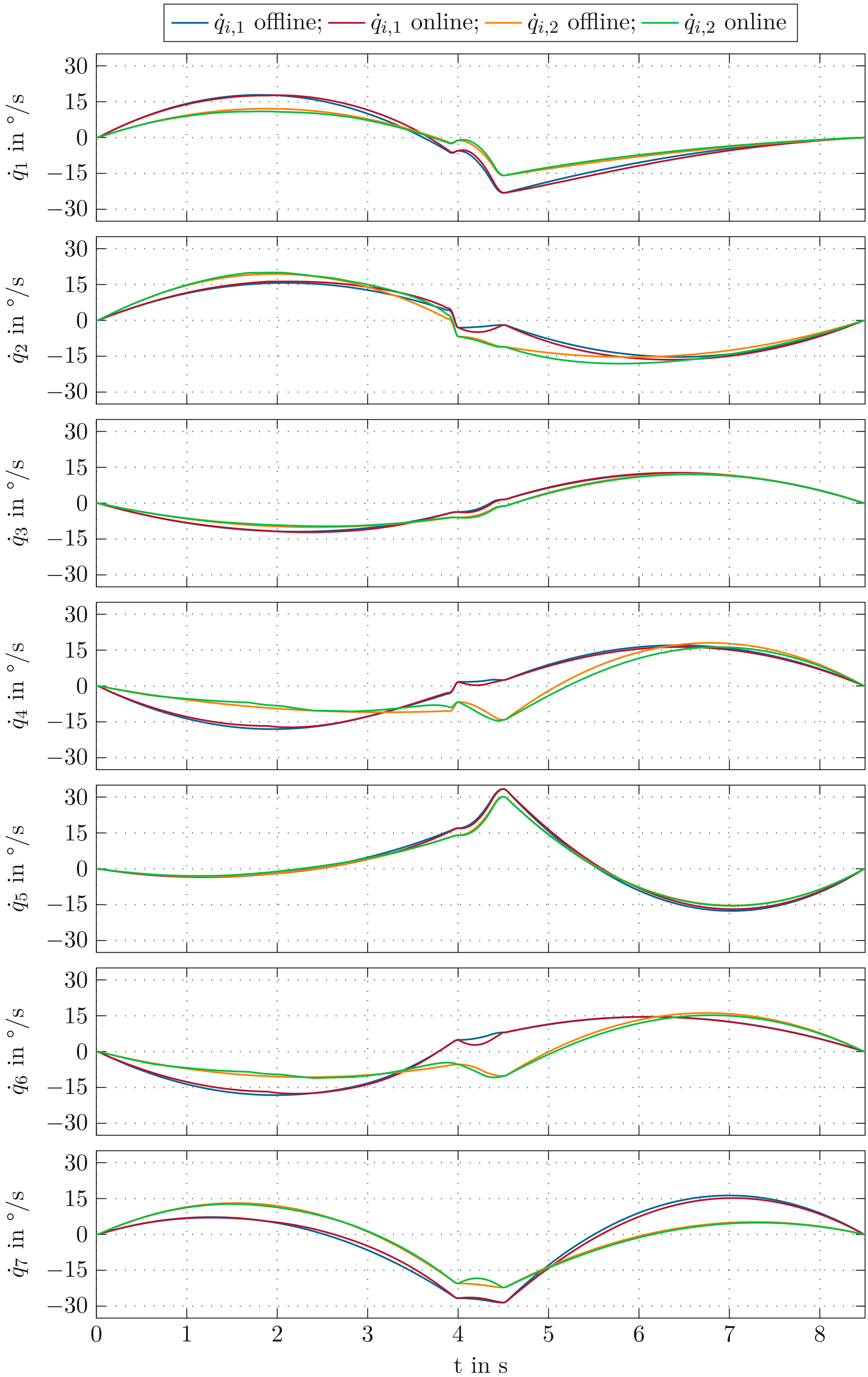


Abbildung 6.8: Interpolierte Teiltrajektorien der Gelenkswinkelgeschwindigkeiten $\dot{q}_1, \dots, \dot{q}_7$ des Roboters mit Initialzustand (6.1a).

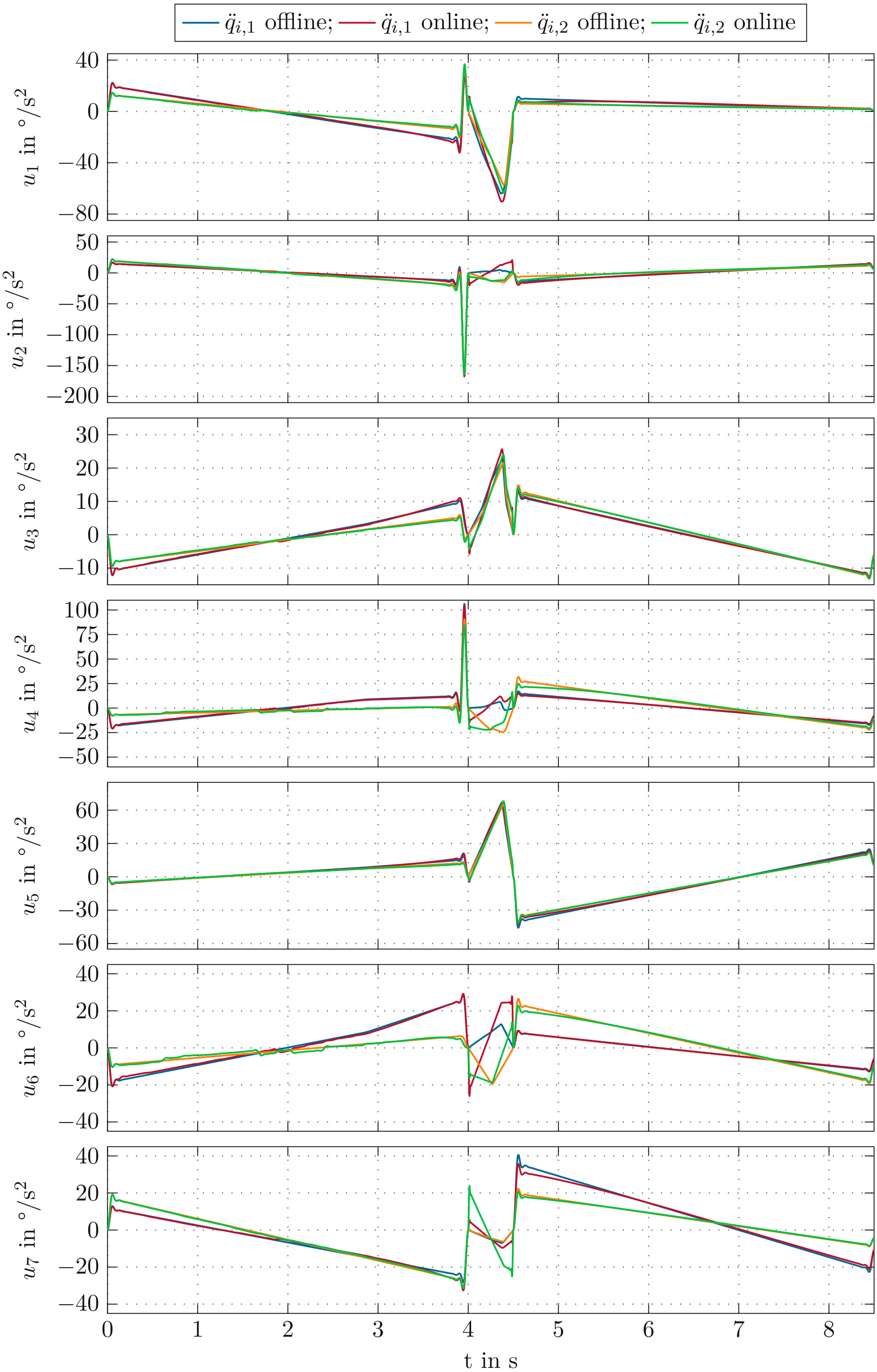


Abbildung 6.9: Interpolierte Teiltrajektorien der Gelenkswinkelbeschleunigungen $\ddot{q}_1, \ldots, \ddot{q}_7$ des Roboters mit Initialzustand (6.1a).

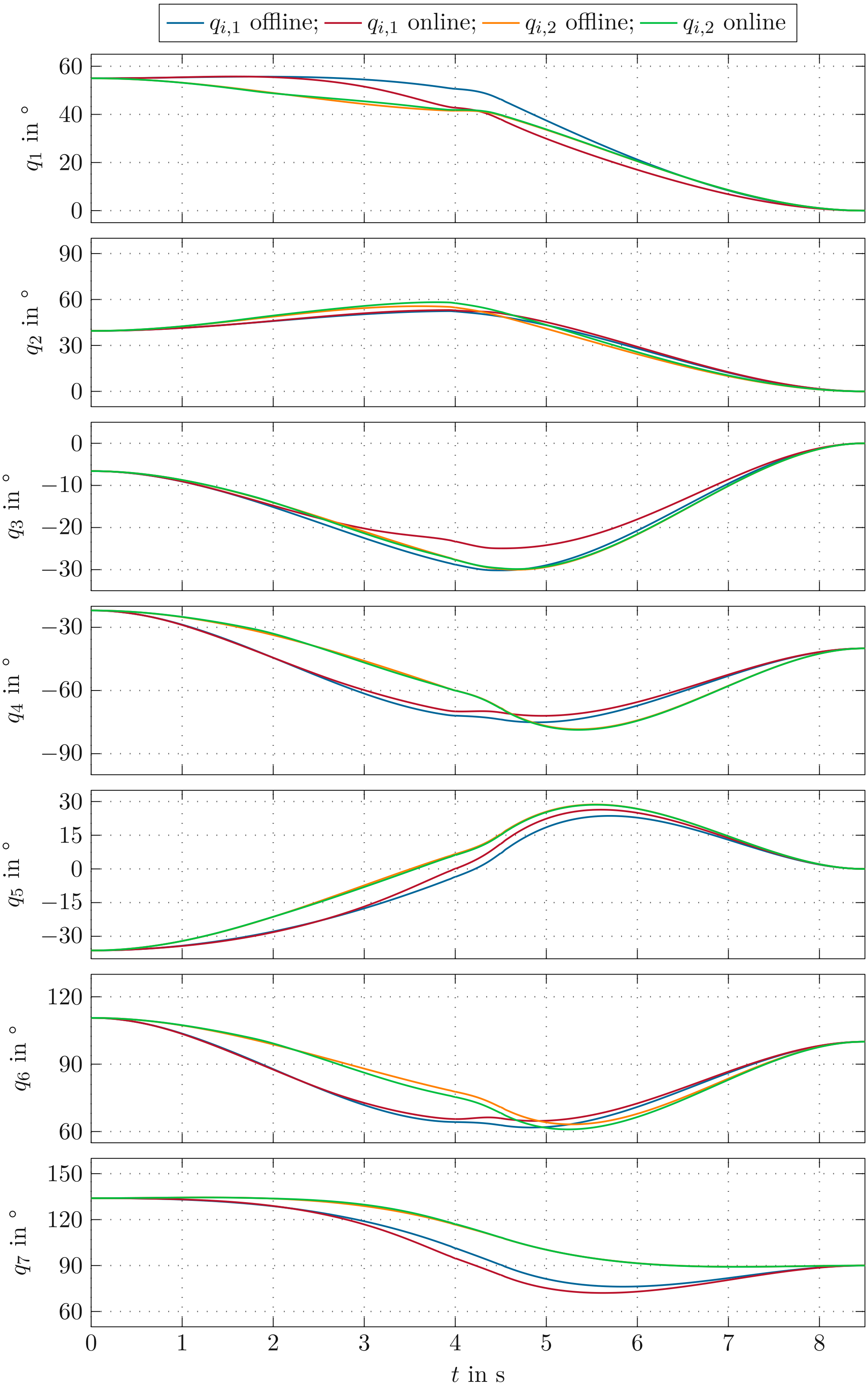


Abbildung 6.10: Interpolierte Teiltrajektorien der Gelenkswinkel $q_1, \ldots, q_7$ des Roboters mit Initialzustand (6.1b).

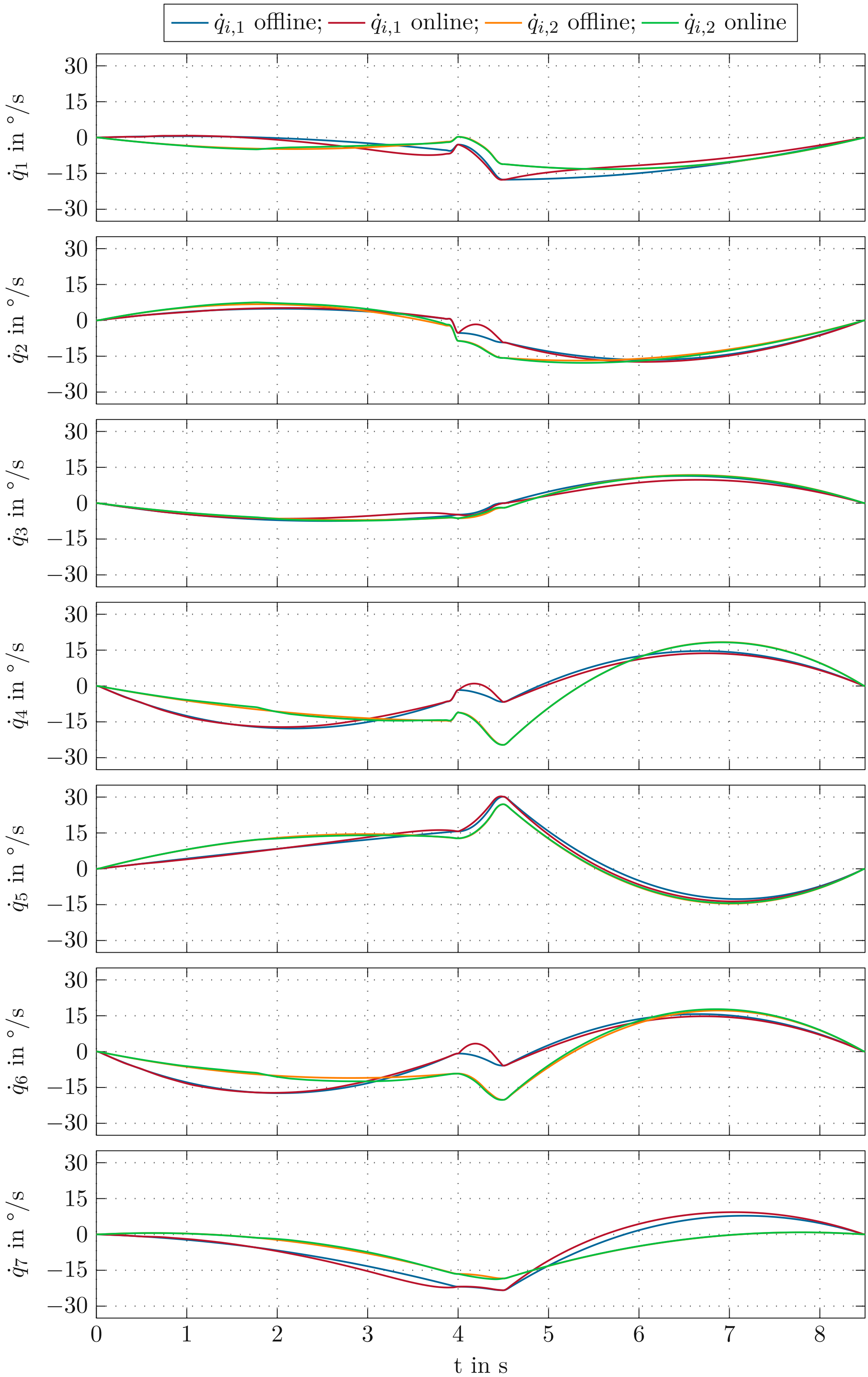


Abbildung 6.11: Interpolierte Teiltrajektorien der Gelenkswinkelgeschwindigkeiten $\dot{q}_1, \ldots, \dot{q}_7$ des Roboters mit Initialzustand (6.1b).

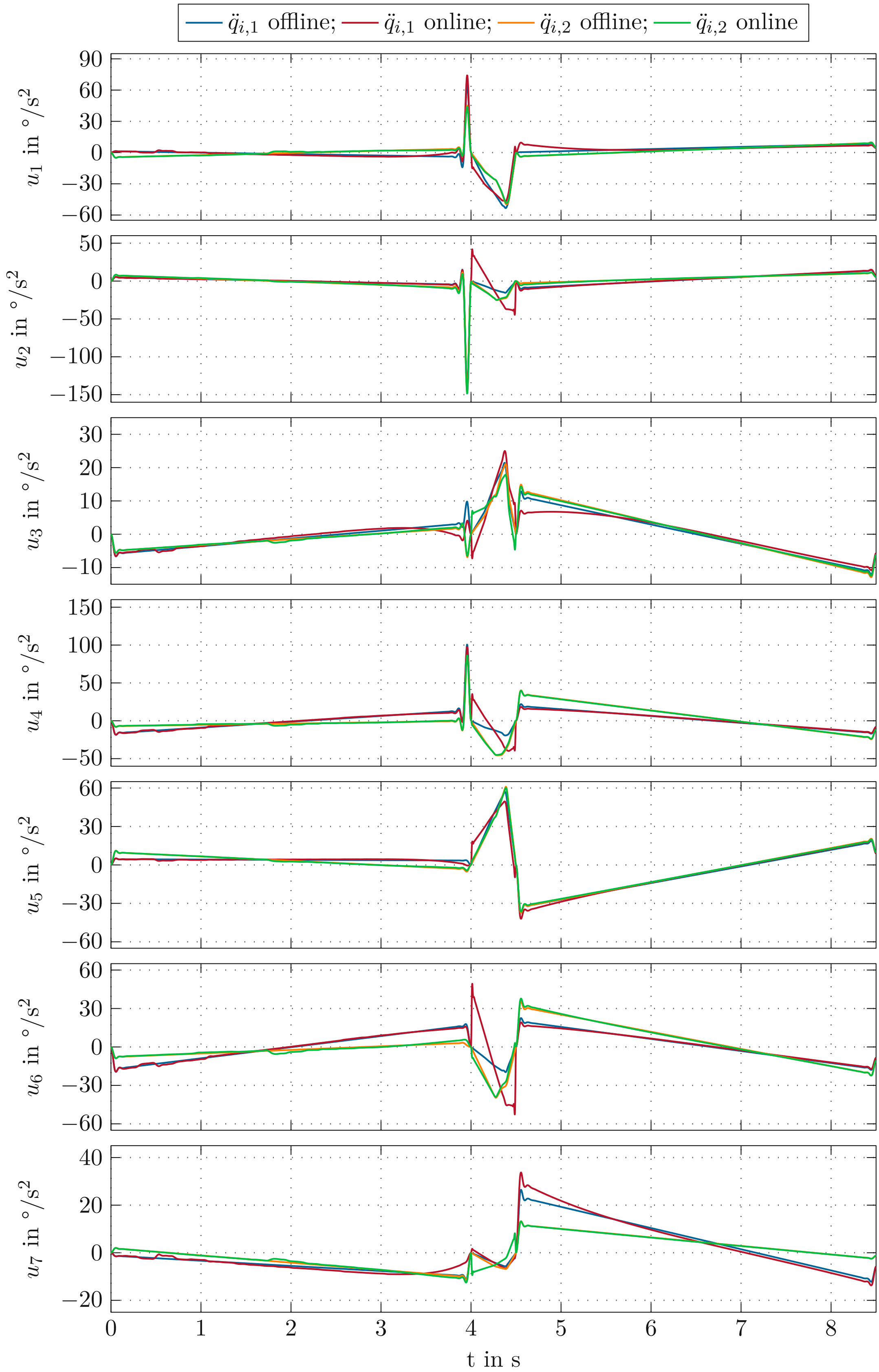
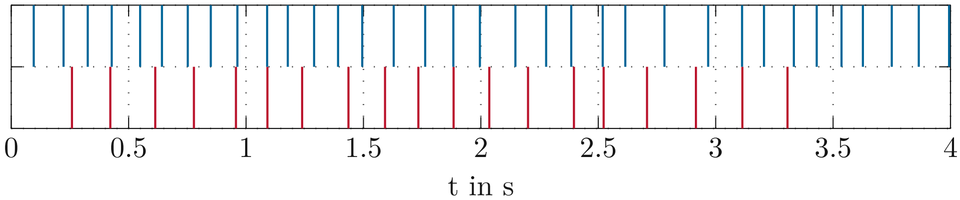
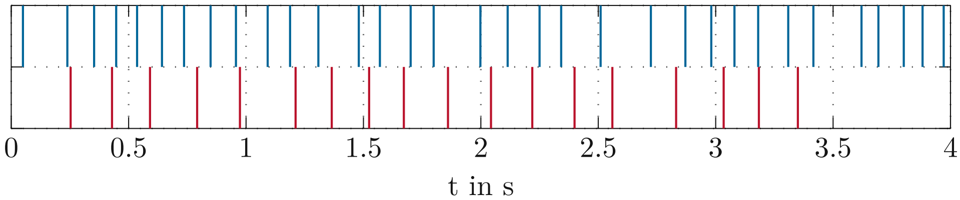


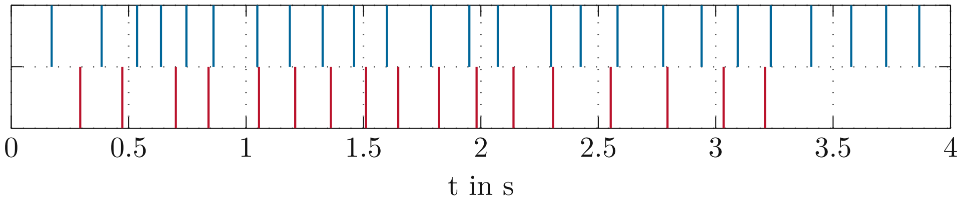
Abbildung 6.12: Interpolierte Teiltrajektorien der Gelenkswinkelbeschleunigungen $\ddot{q}_1, \ldots, \ddot{q}_7$ des Roboters mit Initialzustand (6.1b).



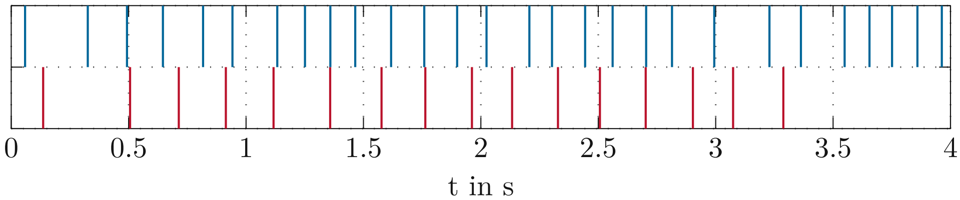
(a) Erster Versuch mit Initialzustand (6.1a).



(b) Zweiter Versuch mit Initialzustand (6.1a).



(c) Erster Versuch mit Initialzustand (6.1b).



(d) Zweiter Versuch mit Initialzustand (6.1b).

Abbildung 6.13: Zeitpunkte der Trajektorienupdates in rot und der Messupdates des Kalman-Filters in blau während der Roboterbewegung von der Initialposition zum Objekt.

6.3 Offline-Trajektorienplanung Monte-Carlo-Simulation

In diesem Abschnitt wird eine Monte-Carlo-Simulation der Offline-Trajektorienplanung durchgeführt. Für eine statistische Auswertung der Erfolgsrate und um mögliche Bereiche mit einer niedrigen Erfolgsrate zu finden, wurde für 20000 zufällige gleichverteilte Objekt-Posen eine Offline-Trajektorienplanung durchgeführt. Der Roboter befindet sich in der Initialposition (6.1a). Die Position des Objekts ist dabei eine gleichverteilte Zufallsvariable $\breve{\mathbf{p}}_0^o = \left[p_{0,x}^o; p_{0,y}^o; p_{0,z}^o\right] \in \left[\underline{\mathbf{p}_0^o}, \overline{\mathbf{p}_0^o}\right]$ mit $\underline{\mathbf{p}_0^o} = [0{,}6\,\mathrm{m}; -0{,}2\,\mathrm{m}; 0{,}15\,\mathrm{m}]$ und $\overline{\mathbf{p}_0^o} = [0{,}75\,\mathrm{m}; 0{,}2\,\mathrm{m}; 0{,}25\,\mathrm{m}]$. Die Orientierung des Objekts ist mit den gleichverteilten Zufallsvariablen der Roll-, Nick- und Gierwinkel $\alpha \in [-2{,}5°; 2{,}5°]$, $\beta = [-15°; 15°]$ und $\gamma = [-2{,}5°; 2{,}5°]$ über

$$\breve{\mathbf{R}}_0^o = \mathbf{R}_{x_o,\frac{\pi}{2}} \mathbf{R}_{y_o,\pi} \mathbf{R}_{z_o,\alpha} \mathbf{R}_{y_o,\beta} \mathbf{R}_{x_o,\gamma} \tag{6.3}$$

festgelegt. Mit der zufälligen Objekt-Pose $\left[\breve{\mathbf{R}}_0^o, \breve{\mathbf{p}}_0^o\right]$ wird eine Offline-Trajektorienplanung nach Abschnitt 4.1 bis 4.3 durchgeführt und anschließend das Status-Bit des Optimierungsalgorithmus für jedes einzelne Optimierungsproblem ausgewertet. Das Status-Bit gibt an, ob eine optimale Lösung gefunden wurde. Die Anzahl der gefundenen optimalen Lösungen sowie die Erfolgsraten sind in Tabelle 6.1 angegeben. Die Auswertung zeigt, dass die Berechnung der Robotertrajektorie während des Schließens der Finger (4.30) und der Roboterbewegung zur Zielposition nach dem Greifen (4.46) die geringste Erfolgsrate haben. Alle anderen Optimierungsprobleme haben eine Erfolgsrate von über 90%. An dieser Stelle sei erwähnt, dass jedes gelieferte Ergebnis des Optimierungsalgorithmus aus (4.30) verwendet wird, um eine optimale Lösung für (4.44) und (4.46) zu finden. Eine Verbesserung der Erfolgsrate von (4.30) könnte somit direkt in eine Verbesserung der Erfolgsrate von (4.44) und (4.46) resultieren.

Optimierungsproblem	(4.7)	(4.17)	(4.19)	(4.30)	(4.44)	(4.46)
Erfolgreich	20000	19598	20000	13440	18780	15845
Erfolgsrate	100%	98%	100%	67,2%	93,9%	79,2%

Tabelle 6.1: Erfolgsrate der einzelnen Optimierungsprobleme der Offline-Trajektorienplanung mit 20000 zufälligen Objekt-Posen.

Um Bereiche der Objektposen $\left[\breve{\mathbf{R}}_0^o, \breve{\mathbf{p}}_0^o\right]$ zu finden, in denen leicht oder kaum eine optimale Trajektorie berechnet werden kann, sind die ersten 10000 Simulationsergebnisse in der sog. Streumatrix in Abbildung 6.14 dargestellt. Dabei wird die Farbe rot verwendet, wenn für jedes einzelne Optimierungsproblem eine optimale Lösung gefunden werden konnte. Liefert mindestens ein Optimierungsproblem nicht die optimale Lösung, ist der jeweilige Punkt in blau eingezeichnet. Auf der Diagonalen der Streumatrix sind die Histogramme der Zufallsvariablen dargestellt. Mit ihnen kann der Einfluss der jeweiligen Zufallsvariablen auf das Lösungsverhalten beurteilt werden. Für die Winkel β und γ zeigt sich z. B. ein besseres Konvergenzverhalten für positive Werte. Während das Konvergenzverhalten für β linear steigt, ist für γ eine sprungförmige Verbesserung um den Nullpunkt zu

beobachten. Mit zunehmender Entfernung des Objekts in Bezug zur Roboterbasis $p^o_{0,x}$ zeigt sich eine leichte Verschlechterung. Das Histogramm der anderen Zufallsvariablen ist annähernd konstant und somit für den simulierten Bereich kaum eine Abhängigkeit auf das Lösungsverhalten erkennbar. Aus den einzelnen Plots in der linken unteren Hälfte lassen sich ebenfalls die Bereiche für eine gute bzw. schlechte Erfolgsrate identifizieren. Jene Bereiche mit einer größeren Dichte an blauen Punkten stellen die Bereiche mit einem schlechten Lösungsverhalten dar. Die rechte obere Hälfte wird nicht weiter kommentiert, da sie die gespiegelte Version der linken unteren Hälfte darstellt.

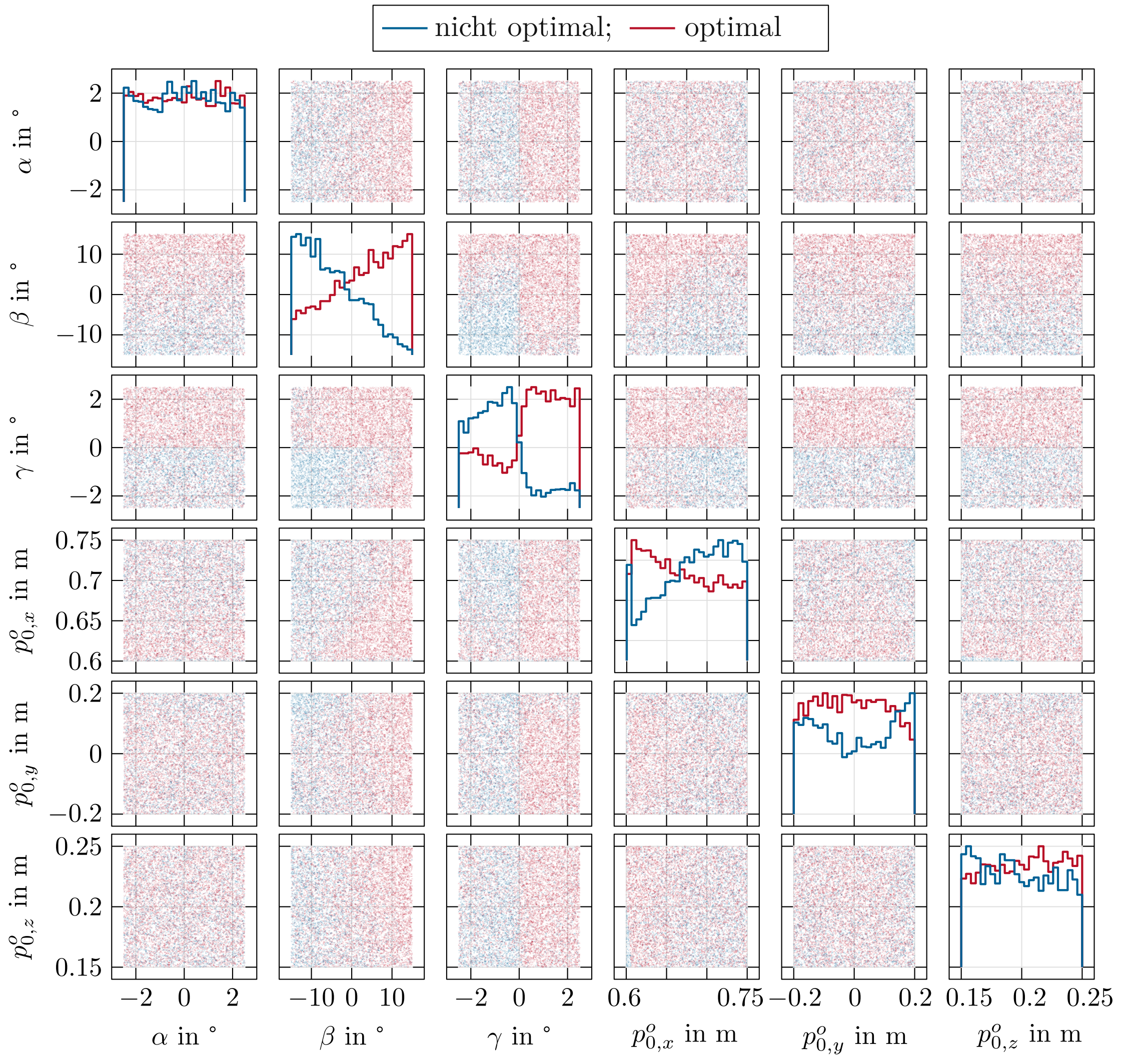


Abbildung 6.14: Streumatrix der Monte-Carlo-Simulation der Offline-Trajektorienplanung.

7 Zusammenfassung

In dieser Arbeit wurde erfolgreich eine Methode für das Greifen eines optisch detektierten Objekts entworfen. Dazu wurde auf einem Industrieroboter eine Greifhand und eine Kamera montiert. Durch eine Objekterkennung wurde das Objekt in den Bildern detektiert und die Lage geschätzt. Für die geschätzte Lage des Objekts wurde offline eine Trajektorie für den Roboter und die Finger der Greifhand geplant. Ziel war es, dass sich der Roboter während des Schließens der Finger kontinuierlich weiterbewegt. Die verwendete Methode der Objekterkennung ist mit Schätzfehlern behaftet. Je geringer die Entfernung zwischen dem Objekt und der Kamera, umso geringer sind die Schätzfehler. Um auf die besser werdenden Schätzungen der Pose während der Greifbewegung zu reagieren, wurde ein schnelles online Update der Robotertrajektorie implementiert.

Im ersten Teil dieser Arbeit wurden die mathematischen Modelle hergeleitet. Da die exakten Masseverteilungen der Greifhand und der Kamera nicht bekannt sind, wurden sie als Punktmasse modelliert und die Parameter geschätzt. Der verwendete Roboter wurde mit den zwei Punktmassen als Endeffektor erweitert und über den Euler-Lagrange Formalismus die Bewegungsgleichungen in der üblichen Form hergeleitet. Für einen erfolgreichen Griff wurden neue Greifzustände eingeführt und eine Vorwärts- bzw. Rückwärtskinematik zum Umrechnen zwischen den Fingerzuständen und den Greifzuständen hergeleitet. Aufgrund der internen Steuerung der Greifhand wurde auf ein dynamisches Modell verzichtet.

Für die Objekterkennung wurden am Objekt sog. ArUco-Marker angebracht. Die Marker können im Bild schnell identifiziert werden. Aus den bekannten Abständen zwischen den Eckpunkten und den Positionen am Objekt wurde die Lage des Objekts in Bezug zur Kamera berechnet. Um den Einfluss etwaiger Ausreißer zu verringern und eine bessere Schätzung zwischen den Kamerabildern zu ermöglichen, wurde ein Kalman-Filter entworfen. Die Eingangsgrößen des Kalman-Filters sind die lineare Geschwindigkeit und die Winkelgeschwindigkeit der Kamera. Mit der geschätzten Pose anhand der ArUco-Marker wurde ein Messupdate des Kalman-Filters durchgeführt.

Durch eine Offline-Trajektorienplanung wurde anhand der geschätzten Pose des Kalman-Filters die Bewegung des Roboters und der Greifhand geplant. Dazu wurde zunächst am Objekt ein Greifpunkt definiert. Die Trajektorie der Finger wurde so geplant, dass sich der Greifpunkt zwischen den Fingern und der des Objekts überlagern und die Oberfläche der Finger tangential zur Objektoberfläche ist. Die Trajektorie des Roboters wurde in drei einzelne Trajektorien unterteilt. In der ersten Teiltrajektorie bewegt sich der Endeffektor von der Initialposition zum Objekt, in der zweiten Teiltrajektorie schließen sich die Finger und in der dritten Teiltrajektorie befindet sich das Objekt an der Greifhand und der Endeffektor bewegt sich zur Zielposition. Für die erste Teiltrajektorie wurden Beschränkungen definiert, damit das Objekt immer im Sichtfeld der Kamera bleibt.

Die geplanten Trajektorien des Roboters und der Greifhand wurden vorgesteuert. Um etwaige Abweichungen gering zu halten bzw. zu eliminieren, wurde jeweils ein Trajek-

torienfolgeregler entworfen. Da die Zentrifugal- und Corioliskräfte bei den auftretenden Geschwindigkeiten nur einen geringen Einfluss haben, wurden sie vernachlässigt.

Die entwickelte Methode wurde in mehreren Versuchen verifiziert und getestet. Während der ersten Teiltrajektorie werden im Schnitt 17 erfolgreiche Updates durchgeführt. Dabei hat sich gezeigt, dass die numerische Berechnung der neuen Zielposition über die inverse Kinematik nicht immer konvergiert. In diesem Fall wurde die Zielposition aus dem vorigen Update übernommen.

A Anhang

A.1 Parameter

A.1.1 Translationen und Rotationen zur Beschreibung der Kinematik

In Tabelle A.1 sind die Rotationsmatrizen und Translationsvektoren zur Beschreibung der Kinematik des Gesamtsystems aus Abschnitt 2 angegeben.

KOS i	KOS j	Translation $\left(\mathbf{d}_i^j\right)^{\mathrm{T}}$	Rotationsmatrix $\mathbf{R}_i^j(\mathbf{q})$
0	1	$[0,0,l_0]$	$\mathbf{R}_{z,q_1}$
1	2	$[0,-d_1,l_1]$	$\mathbf{R}_{x,-\frac{\pi}{2}}\mathbf{R}_{z,q_2}$
2	3	$[0,-l_2,d_1]$	$\mathbf{R}_{x,\frac{\pi}{2}}\mathbf{R}_{z,q_3}$
3	4	$[0,d_3,l_3]$	$\mathbf{R}_{x,\frac{\pi}{2}}\mathbf{R}_{z,q_4}$
4	5	$[0,l_4,d_3]$	$\mathbf{R}_{x,-\frac{\pi}{2}}\mathbf{R}_{z,q_5}$
5	6	$[0,-d_5,l_5]$	$\mathbf{R}_{x,-\frac{\pi}{2}}\mathbf{R}_{z,q_6}$
6	7	$[0,-l_6,d_5]$	$\mathbf{R}_{x,\frac{\pi}{2}}\mathbf{R}_{z,q_7}$
7	e	$[0,0,l_7]$	$\mathbf{E}_{3\times 3}$
7	c	$[0,d_c,l_7+l_c]$	$\mathbf{R}_{z,\pi}\mathbf{R}_{x,\theta_c}$
7	h	$[0,0,l_7+l_h]$	$\mathbf{R}_{z,\frac{\pi}{2}}$
1	$b1$	$[s_{l1x},s_{l1y},s_{l1z}]$	$\mathbf{E}_{3\times 3}$
2	$b2$	$[s_{l2x},s_{l2y},s_{l2z}]$	$\mathbf{E}_{3\times 3}$
3	$b3$	$[s_{l3x},s_{l3y},s_{l3z}]$	$\mathbf{E}_{3\times 3}$
4	$b4$	$[s_{l4x},s_{l4y},s_{l4z}]$	$\mathbf{E}_{3\times 3}$
5	$b5$	$[s_{l5x},s_{l5y},s_{l5z}]$	$\mathbf{E}_{3\times 3}$
6	$b6$	$[s_{l6x},s_{l6y},s_{l6z}]$	$\mathbf{E}_{3\times 3}$
7	$b7$	$[s_{l7x},s_{l7y},s_{l7z}]$	$\mathbf{E}_{3\times 3}$
c	bc	$[s_{cx},s_{cy},s_{cz}]$	$\mathbf{E}_{3\times 3}$
h	bh	$[s_{hx},s_{hy},s_{hz}]$	$\mathbf{E}_{3\times 3}$

Tabelle A.1: Translation und Rotation zwischen den Koordinatensystemen $(0_i x_i y_i z_i)$ und $(0_j x_j y_j z_j)$ für die Beschreibung der Kinematik des Gesamtsystems.

A.1.2 Roboter KUKA LBR iiwa R14 820

Die Parameter des Roboters aus Abbildung 2.1 sind in Tabelle A.2 angegeben.

Parameter	Value	
l_0	$152{,}5 \cdot 10^{-3}$	m
l_1	$207{,}5 \cdot 10^{-3}$	m
l_2	$232{,}5 \cdot 10^{-3}$	m
l_3	$187{,}5 \cdot 10^{-3}$	m
l_4	$212{,}5 \cdot 10^{-3}$	m
l_5	$187{,}5 \cdot 10^{-3}$	m
l_6	$79{,}6 \cdot 10^{-3}$	m
l_7	$72{,}4 \cdot 10^{-3}$	m
l_c	$69{,}04 \cdot 10^{-3}$	m
l_h	$104{,}3 \cdot 10^{-3}$	m
d_1	$13 \cdot 10^{-3}$	m
d_3	$11 \cdot 10^{-3}$	m
d_5	$62 \cdot 10^{-3}$	m
d_c	$135 \cdot 10^{-3}$	m
m_{b1}	6,495	kg
m_{b2}	8,807	kg
m_{b3}	2,8	kg
m_{b4}	5,283	kg
m_{b5}	1,889	kg
m_{b6}	2,32	kg
m_{b7}	0,863	kg
m_c	0,63	kg
m_h	2,49	kg

Parameter	Value	
$I_{b1,xx}$	$69{,}076 \cdot 10^{-3}$	kg m^2
$I_{b1,yy}$	$71{,}108 \cdot 10^{-3}$	kg m^2
$I_{b1,zz}$	$20{,}062 \cdot 10^{-3}$	kg m^2
$I_{b2,xx}$	$82{,}470 \cdot 10^{-3}$	kg m^2
$I_{b2,yy}$	$16{,}411 \cdot 10^{-3}$	kg m^2
$I_{b2,zz}$	$87{,}351 \cdot 10^{-3}$	kg m^2
$I_{b3,xx}$	$23{,}015 \cdot 10^{-3}$	kg m^2
$I_{b3,yy}$	$22{,}866 \cdot 10^{-3}$	kg m^2
$I_{b3,zz}$	$55{,}448 \cdot 10^{-3}$	kg m^2
$I_{b4,xx}$	$47{,}278 \cdot 10^{-3}$	kg m^2
$I_{b4,yy}$	$9{,}647 \cdot 10^{-3}$	kg m^2
$I_{b4,zz}$	$46{,}652 \cdot 10^{-3}$	kg m^2
$I_{b5,xx}$	$13{,}835 \cdot 10^{-3}$	kg m^2
$I_{b5,yy}$	$11{,}685 \cdot 10^{-3}$	kg m^2
$I_{b5,zz}$	$5{,}718 \cdot 10^{-3}$	kg m^2
$I_{b6,xx}$	$7{,}327 \cdot 10^{-3}$	kg m^2
$I_{b6,yy}$	$4{,}776 \cdot 10^{-3}$	kg m^2
$I_{b6,zz}$	$6{,}067 \cdot 10^{-3}$	kg m^2
$I_{b7,xx}$	$0{,}884 \cdot 10^{-3}$	kg m^2
$I_{b7,yy}$	$0{,}888 \cdot 10^{-3}$	kg m^2
$I_{b7,zz}$	$1{,}105 \cdot 10^{-3}$	kg m^2
$I_{bc,xx}$	$1{,}662 \cdot 10^{-3}$	kg m^2
$I_{bc,yy}$	$0{,}823 \cdot 10^{-3}$	kg m^2
$I_{bc,zz}$	$1{,}317 \cdot 10^{-3}$	kg m^2
$I_{bh,xx}$	$8{,}200 \cdot 10^{-3}$	kg m^2
$I_{bh,yy}$	$8{,}254 \cdot 10^{-3}$	kg m^2
$I_{bh,zz}$	$3{,}503 \cdot 10^{-3}$	kg m^2

Tabelle A.2: Parameter des KUKA LBR iiwa 14 R820 Roboters.

A.1.3 Greifhand Schunk SDH2

Die Parameter der Schunk Greifhand SDH2 aus Abbildung 2.3 sind in Tabelle A.3 angegeben.

Parameter	Wert	
h_1	$41{,}5 \cdot 10^{-3}$	m
b_1	$33 \cdot 10^{-3}$	m
l_1	$19{,}05 \cdot 10^{-3}$	m
h_2	$86{,}5 \cdot 10^{-3}$	m
b_M	$15{,}29 \cdot 10^{-3}$	m
b_{T1}	$14{,}82 \cdot 10^{-3}$	m
b_{T2}	$5{,}15 \cdot 10^{-3}$	m
h_{T1}	$18{,}51 \cdot 10^{-3}$	m
h_{T2}	$33{,}26 \cdot 10^{-3}$	m
h_{T3}	$47{,}5 \cdot 10^{-3}$	m
h_{T4}	$66{,}06 \cdot 10^{-3}$	m
r_{T1}	$60 \cdot 10^{-3}$	m
r_{T2}	$61 \cdot 10^{-3}$	m
s_T	$66{,}26 \cdot 10^{-3}$	m
s_C	$56{,}75 \cdot 10^{-3}$	m
θ	54,42	°
η	4,46	°
$q_{h1,0}$	90	°
$q_{h2,0}$	-10	°
$q_{h3,0}$	0	°
$q_{h4,0}$	90	°
$q_{h5,0}$	60	°
$q_{h6,0}$	-10	°
$q_{h7,0}$	0	°

Tabelle A.3: Parameter der Greifhand Schunk SDH2.

A.1.4 Kalibrierung der Kamera

Eine Kalibrierung der Kamera mit Hilfe eines Schachbrettmusters liefert die in Tabelle A.4 angegebenen intrinsischen Parameter der Kamera.

A.1.5 Kalman-Filter

Die verwendeten Kovarianzmatrizen des Kalman-Filters sind in Tabelle A.5 angegeben. Die Werte für $\mathbf{R}_k$ sind aus den Rauscheigenschaften der Posenschätzung des Objekts abgeleitet. Um die Auswirkung von Ausreißern gering zu halten, wurden die Werte für $\mathbf{Q}_k$ manuell gewählt.

Parameter	Wert
f_x	1122,98
f_y	1122,28
c_x	513,52
c_y	507,50

Tabelle A.4: Intrinsischen Parameter der Kamera Basler avA1000-100gc.

Parameter	Wert
$\mathbf{R}_k$	diag ([0,0004; 0,0004; 0,0004; 0,0004; 0,0004; 0,0004; 0,0004])
$\mathbf{Q}_k$	diag ([2,5; 2,5; 2,5; 10; 10; 10])

Tabelle A.5: Kovarianzmatrizen des Kalman-Filters.

A.1.6 Reglerparameter

In diesem Abschnitt sind die Parameter der Trajektorienfolgeregler des Roboters in Tabelle A.6 und der Greifhand in Tabelle A.7 angegeben.

Parameter	Wert
$\mathbf{K}_0$	diag ([1200; 1200; 1200; 1200; 1200; 1200; 1200])
$\mathbf{K}_1$	diag ([60; 60; 60; 60; 60; 60; 60])
$\mathbf{K}_I$	diag ([8000; 8000; 8000; 8000; 8000; 8000; 8000])
$\mathbf{K}_\tau$	diag ([4; 4; 4; 5; 3; 2,5; 2,5])
$\mathbf{D}_\tau$	diag ([0,015; 0,015; 0,015; 0,02; 0,01; 0,01; 0,01])
$\mathbf{B}$	diag ([10,83; 10,83; 4,89; 4,89; 1,47; 0,47; 0,47])

Tabelle A.6: Parameter des Trajektorienfolgereglers des Roboters.

A.1.7 Beschränkungen

Die gültigen Beschränkungen aller Größen sind in Tabelle A.8 angegeben.

Parameter	Wert
$\mathbf{K}_P$	diag ([2; 2; 2; 2; 2; 2; 2])
$\mathbf{K}_I$	diag ([0,5; 0,5; 0,5; 0,5; 0,5; 0,5; 0,5])

Tabelle A.7: Parameter des Reglers der Greifhand.

Par.	Wert
$\underline{\mathbf{q}_G}$	[0 mm; −10°; −30 mm; 160 mm]
$\overline{\mathbf{q}_G}$	[80 mm; 10°; 30 mm; 170 mm]
$\underline{\mathbf{q}_h}$	[0°; −90°; −90°; −90°; −90°; −90°; −90°]
$\overline{\mathbf{q}_h}$	[90°; 90°; 90°; 90°; 90°; 90°; 90°]
$\underline{\dot{\mathbf{q}}_h}$	[−80 °/s; −100 °/s; −100 °/s; −100 °/s; −100 °/s; −100 °/s; −100 °/s]
$\overline{\dot{\mathbf{q}}_h}$	[80 °/s; 100 °/s; 100 °/s; 100 °/s; 100 °/s; 100 °/s; 100 °/s]
$\underline{\ddot{\mathbf{q}}_h}$	1000·[−5 °/s^2; −0,4 °/s^2; −1,5 °/s^2; −0,4 °/s^2; −1,5 °/s^2; −0,4 °/s^2; −1,5 °/s^2]
$\overline{\ddot{\mathbf{q}}_h}$	1000·[5 °/s^2; 0,4 °/s^2; 1,5 °/s^2; 0,4 °/s^2; 1,5 °/s^2; 0,4 °/s^2; 1,5 °/s^2]
$\underline{\mathbf{q}}$	[−170°; −120°; −170°; −120°; −170°; −120°; −175°]
$\overline{\mathbf{q}}$	[170°; 120°; 170°; 120°; 170°; 120°; 175°]
$\underline{\dot{\mathbf{q}}}$	0,4·[−85 °/s; −85 °/s; −100 °/s; −75 °/s; −130 °/s; −135 °/s; −135 °/s]
$\overline{\dot{\mathbf{q}}}$	0,4·[85 °/s; 85 °/s; 100 °/s; 75 °/s; 130 °/s; 135 °/s; 135 °/s]
$\underline{\boldsymbol{\tau}}$	0,4·[320 N m; 320 N m; 176 N m; 176 N m; 110 N m; 40 N m; 40 N m]
$\overline{\boldsymbol{\tau}}$	0,4·[320 N m; 320 N m; 176 N m; 176 N m; 110 N m; 40 N m; 40 N m]
$\underline{\mathbf{C}}$	[−8,5 N m; −10 N m; −5 N m; −5,7 N m; −1,5 N m; −1,8 N m; −0,25 N m]
$\overline{\mathbf{C}}$	[8,5 N m; 10 N m; 5 N m; 5,7 N m; 1,5 N m; 1,8 N m; 0,25 N m]
$\underline{T_1}$	4 s
$\overline{T_1}$	7 s
$\underline{T_3}$	4 s
$\overline{T_3}$	7 s
$\underline{\tilde{\mathbf{p}}_c^o}$	[0 px; 0 px]
$\overline{\tilde{\mathbf{p}}_c^o}$	[1024 px; 1024 px]
$\underline{\Delta\tilde{\mathbf{p}}_c^o}$	[−50 px; −50 px]
$\overline{\Delta\tilde{\mathbf{p}}_c^o}$	[50 px; 50 px]
$\underline{\Delta\mathbf{d}_0^G}$	[−0,5 mm; −0,5 mm; −0,5 mm]
$\overline{\Delta\mathbf{d}_0^G}$	[0,5 mm; 0,5 mm; 0,5 mm]
$\underline{\Delta\mathbf{d}_0^h}$	[−0,5 mm; −0,5 mm; −0,5 mm]
$\overline{\Delta\mathbf{d}_0^h}$	[0,5 mm; 0,5 mm; 0,5 mm]

Tabelle A.8: Beschränkungen

Literatur

[1] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox und S. Birchfield, „Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects“, *CoRR*, Jg. abs/1809.10790, 2018. arXiv: `1809.10790`. Adresse: `http://arxiv.org/abs/1809.10790`.

[2] V. Lepetit, F. Moreno-Noguer und P. Fua, „EPnP: An accurate O(n) solution to the PnP problem“, *International Journal of Computer Vision*, Jg. 81, Nr. 2, S. 155–166, 2009.

[3] T. Do, M. Cai, T. Pham und I. D. Reid, „Deep-6DPose: Recovering 6D Object Pose from a Single RGB Image“, *CoRR*, Jg. abs/1802.10367, 2018. arXiv: `1802.10367`. Adresse: `http://arxiv.org/abs/1802.10367`.

[4] S. Peng, Y. Liu, Q. Huang, H. Bao und X. Zhou, „PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation“, *CoRR*, Jg. abs/1812.11788, 2018. arXiv: `1812.11788`. Adresse: `http://arxiv.org/abs/1812.11788`.

[5] B. Tekin, S. N. Sinha und P. Fua, „Real-Time Seamless Single Shot 6D Object Pose Prediction“, *CoRR*, Jg. abs/1711.08848, 2017. arXiv: `1711.08848`. Adresse: `http://arxiv.org/abs/1711.08848`.

[6] T. To, J. Tremblay, D. McKay, Y. Yamaguchi, K. Leung, A. Balanon, J. Cheng, W. Hodge und S. Birchfield, *NDDS: NVIDIA Deep Learning Dataset Synthesizer*, `https://github.com/NVIDIA/Dataset_Synthesizer`, 2018.

[7] G. Wan, „Visual Positioning and Grasping Application of Industrial Robot for Casting Parts“, *Automation, Control and Intelligent Systems*, Jg. 7, Nr. 1, S. 18–24, 2019.

[8] C.-H. G. Li und Y.-M. Chang, „Automated visual positioning and precision placement of a workpiece using deep learning“, *The International Journal of Advanced Manufacturing Technology*, Jg. 104, Nr. 9, S. 4527–4538, 2019.

[9] J. Rauer, W. Wöber und M. Aburaia, „An Autonomous Mobile Handling Robot Using Object Recognition“, Techn. Ber., 2019.

[10] M. N. Vu, C. Hartl-Nesic und A. Kugi, „Fast Swing-Up Trajectory Optimization for a Spherical Pendulum on a 7-DoF Collaborative Robot“, in *2021 IEEE International Conference on Robotics and Automation (ICRA) [Accepted]*, 2021.

[11] M. Vu, P. Zips, A. Lobe, F. Beck, W. Kemmetmüller und A. Kugi, „Fast motion planning for a laboratory 3D gantry crane in the presence of obstacles“, *IFAC-PapersOnLine*, Jg. 53, Nr. 2, S. 9508–9514, 2020.

[12] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*. Berlin, Heidelberg: Springer, 2008, ISBN: 3540692533.

[13] M. A. Roa und R. Suarez, „Grasp Quality Measures: Review and Performance“, *Autonomous Robots*, Jg. 38, Nr. 1, S. 65–88, 2015.

[14] A. Sahbani, S. El-Khoury und P. Bidaud, „An Overview of 3D Object Grasp Synthesis Algorithms“, *Robotics and Autonomous Systems*, Jg. 60, Nr. 3, S. 326–336, 2012.

[15] B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo, *Robotics: Modelling, Planning and Control*, Ser. Advanced Textbooks in Control and Signal Processing. London: Springer, 2010, ISBN: 9781846286414. Adresse: `https://books.google.de/books?id=jPCAFmE-logC`.

[16] D. A. Forsyth und J. Ponce, „Computer vision : a modern approach“, Boston[u.a.]: Pearson, 2012, ISBN: 013608592X.

[17] F. Königseder, W. Kemmetmüller und A. Kugi, „Attitude control strategy for a camera stabilization platform“, *Mechatronics*, Jg. 46, S. 60–69, 2017.

[18] F. L. Markley und J. L. Crassidis, *Fundamentals of spacecraft attitude determination and control*, Ser. Space technology library. New York: Springer, 2014. Adresse: `http://cds.cern.ch/record/1707461`.

[19] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Cambridge University Press, 2009, ISBN: 0898716888.

[20] L. Piegl und W. Tiller, *The NURBS Book*. Berlin, Heidelberg: Springer, 1995, ISBN: 3540550690.

[21] S. Buss, „Introduction to inverse kinematics with Jacobian transpose, pseutoinverse and damped least squares methods“, Techn. Ber., 2004.

[22] Schunk. „Montage- und Betriebsanleitung SDH2“. (2018), Adresse: `https://schunk.com/fileadmin/pim/docs/IM0023479.PDF` (besucht am 27. 04. 2021).

[23] basler. „The official python wrapper for the Basler pylon Camera Software Suite.“ (2020), Adresse: `https://github.com/basler/pypylon` (besucht am 27. 04. 2021).

[24] OpenCV. „OpenCV: Open Source Computer Vision Library“. (2019), Adresse: `https://github.com/opencv/opencv/releases/tag/4.2.0` (besucht am 27. 04. 2021).

[25] ——, „Repository for OpenCV’s extra modules“. (2019), Adresse: `https://github.com/opencv/opencv_contrib/releases/tag/4.2.0` (besucht am 27. 04. 2021).

[26] pyads. „Python wrapper for TwinCAT ADS library.“ (2020), Adresse: `https://pypi.org/project/pyads/` (besucht am 27. 04. 2021).

[27] A. Wächter und L. T. Biegler, „On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming“, *Math. Program.*, Jg. 106, Nr. 1, S. 25–57, 2006.

[28] HSL. „A collection of Fortran codes for large scale scientific computation.“ (2018), Adresse: `http://www.hsl.rl.ac.uk/` (besucht am 27. 04. 2021).

[29] I. S. Duff, „MA57—a Code for the Solution of Sparse Symmetric Definite and Indefinite Systems“, *ACM Trans. Math. Softw.*, Jg. 30, Nr. 2, S. 118–144, Juni 2004. Adresse: `https://doi.org/10.1145/992200.992202`.

[30] J. Andersson, J. Gillis, G. Horn, J. Rawlings und M. Diehl, „CasADi: a software framework for nonlinear optimization and optimal control“, *Mathematical Programming Computation*, Jg. 11, S. 1–36, 2019.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct – Regeln zur Sicherung guter wissenschaftlicher Praxis (in der aktuellen Fassung des jeweiligen Mitteilungsblattes der TU Wien), insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In– noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, Juni 2021

Grander Florian